Using PageRank Algorithm in Analyzing Dictionary Graphs and PageRank in Dynamic Graphs

Asefeh Salarinezhad

Mathematics

A thesis submitted in partial fulfilment
of the requirements for the degree of

Masters of Science in Mathematics

Faculty of Mathematics and Science, Brock University
St. Catharines, Ontario

Abstract

In this thesis we are going to analyze the dictionary graphs and some other kinds of graphs using the *PagerRank* algorithm.

We calculated the correlation between the degree and *PageRank* of all nodes for a graph obtained from Merriam-Webster dictionary [Gutenberg (1996)], a French dictionary [Project (1835)] and WordNet [WordNet (2014)] hypernym and synonym dictionaries.

Our conclusion was that *PageRank* can be a good tool to compare the quality of dictionaries.

We studied some artificial social and random graphs. We found that when we omitted some random nodes from each of the graphs, we have not noticed any significant changes in the ranking of the nodes according to their *PageRank*.

We also discovered that some social graphs selected for our study were less resistant to the changes of *PageRank*.

Acknowledgments

First and foremost, I want to thank specially my supervisors, Dr. Henryk Fukś and Dr. Babak Farzad for their support and valuable guides.

Besides my advisors, I would like to thank Dr. Kihel for letting me have him in my thesis committee.

My sincere thanks also goes to the chair of Mathematics and Statistics Department and the rest staff of the department for their insightful support and encouragement.

I thank Brock University which provided me an opportunity to join them as a graduate student.

Last but not the least, I would like to thank my family for supporting me spiritually throughout writing this thesis and my life in general.

Contents

List of Figures

Chapter 1

Introduction

As a result of the digital revolution and advancements of information technology, increasing proportion of activities of any organization in industry and business is now done electronically. The same trend can be observed in everyday life. This phenomenon has created an urgent need for new techniques and tools that can assist us in analyzing the vast amounts of data quickly and reliably.

In many applications, available data are organized as a system of interconnected components or nodes. Various tools for analyzing and comparing such data sets have been developed in recent years, and among them the so-called *PageRank* algorithm plays a prominent role. The *PageRank* algorithm is a useful tool in determining the importance of nodes, and for this reason is has found a number of applications in various field, ranging from internet search engines to purely mathematical problems. The motivation for this work is to study the possibility of applications of *PageRanks* to some novel areas as well as to investigate robustness of ranking produced by this algorithm.

Chapter 2

Preliminaries and notations

2.1 Definitions

2.1.1 Graph

Let us denote a *graph* by G. This graph consists of a non-empty set of vertices and a set of edges which connect the pairs of vertices together. We call the node set $V(G)$ and the edge set $E(G)$. In real world situations, most of the times we use the word network for a graph.

In these situations the graph is a network of items, which we call vertices or nodes. The edges are the links and connections between these items. We can find a lot of examples of systems taking the form of networks in the real world [Newman (2003)].

2.1.2 Directed Graph

A *digraph* or a *directed graph*, let us call it D, is a graph which has a non-empty set of vertices, to be called $V(D)$, and a set of directed arcs or

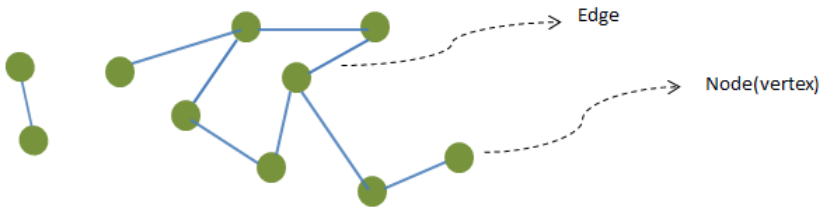

Figure 2.1: *An example graph with ten vertices and 9 edges.*

edges which is a set of ordered pairs of distinct vertices. We call this edge set $A(D)$ [Bang-Jensen and Gutin (2002)]. We define (u, v) as a representative of an ordered pair of nodes. (u, v) is a directed arc, or a directed edge. The graphical form of (u, v) is an arrow drawn between the two vertices. Considering the order of the nodes, the first vertex is called the initial vertex or tail and the second node is known as the terminal vertex or head (because it appears at the arrow head) [Bondy and Murty (1976)].

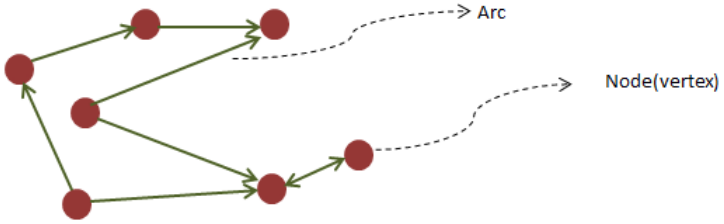

Figure 2.2: *A digraph D.*

2.1.3 Weighted Graph

Some graphs are weighted, which means that their edges are weighted. The weight of an edge is a numerical value associated to each edge of the graph. This weight sometimes is referred to as "cost" of the edge. Most of the times

the weight of an edge is a positive integer. Depending on applications, the weight can be a measure of the length of a route, the load or the capacity of a line, the volume of traffic between locations along a route, etc [McQuain (2010)].

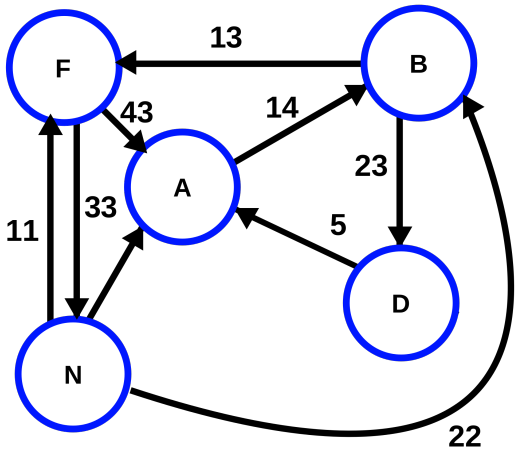

Figure 2.3: *A weighted graph [Galleryhip (2015)].*

2.1.4 Degree of nodes

For a node v, the in-degree of v is the number of directed edges aimed to v and the out-degree of v is the number of directed edges which leave this node [Harris et al. (2008)].

The *degree* $d_G(v) = d(v)$ of a vertex v is the number of edges which are incident to v. The degree is also referred to as valency [Diestel (2010)].

2.1.5 Web Graph

The *web graph* is a graph, to be called W; in which the vertices represent the web pages, and the edges represent the links between pages.

This massive and evolving graph has the following properties: It is a sparse network, has the "small world" topology, and its degree distribution approximately follows a power law [Bonato (2008)].

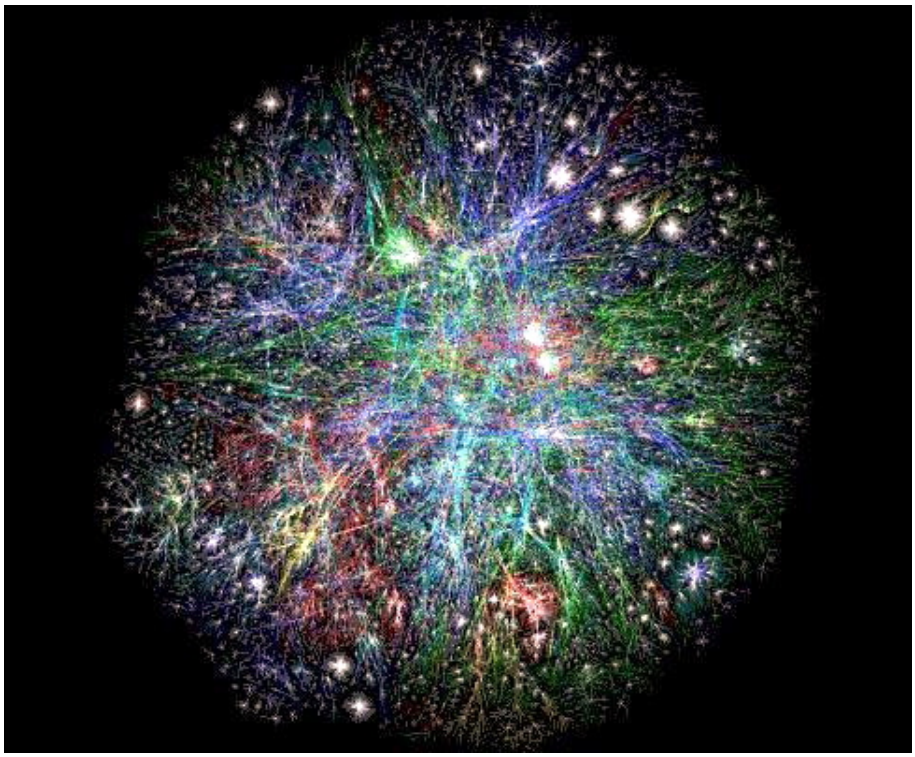

Figure 2.4: *Graph of the World Wide Web [Institute (2015)].*

2.1.6 *PageRank*

PageRank is a well-known algorithm which is used by Google Search as a tool of ranking websites in their search engine results. *PageRank* was named after Larry Page, one of the founders of Google.

PageRank is a method which helps us to measure the importance of website pages, but it can also be used as a tool to rank the nodes in any network.

According to Google, *PageRank* works by considering the number and quality of links which aim to a page to define a rough estimate of how important the website is. It uses the underlying assumption that more important websites (nodes) are likely to receive more links from other websites (nodes) [Wikipedia (2014b)].

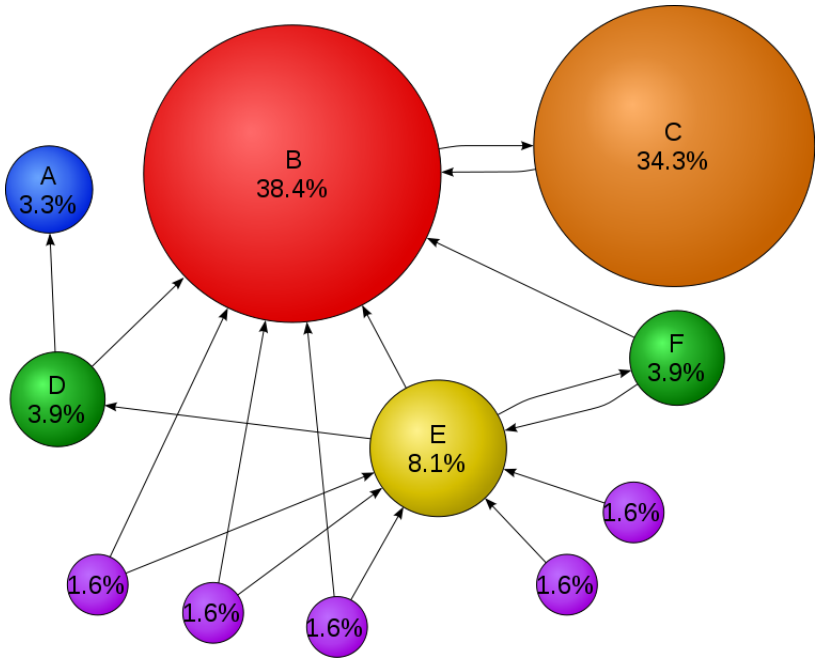

Figure 2.5: *Example of PageRank applied to a simple network. PageRank are expressed as percentage (Google uses a logarithmic scale). Page C has a higher PageRank than Page E, even though there are fewer links to C; the one link to C comes from an important page and hence it is of high value [Wikipedia (2014b)]. In next sessions we describe how to compute the PageRank of these pages.*

Ranking produced by the *PageRank* algorithm can also be understood as a stationary distribution of a random walk on a directed graph. In other words, the *PageRank* of a web page is a value which shows the probability that, at any given moment, a random surfer is visiting this page.

PageRank ranks nodes of a graph G according to the structure of the incoming links.

PageRank Computing

Let us start with a quick description of *PageRank* computing [Ipsen and Wills (2005)] and some issues in computing it.

If π_i is the importance score of node i, finding the *PageRank* vector means finding these π_i values. For a graph with n nodes, they will be written as a vector,

$$\begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_n \end{bmatrix}.$$

2.1.6.1 Normalized *PageRank*

According to definition of *PageRank*, the importance of every node is, roughly speaking, equal to the sum of importance of its neighbors which are aim at it. So, conceptually in Figure 2.6,

$$\pi_3 = \pi_1 + \pi_2. \tag{2.1}$$

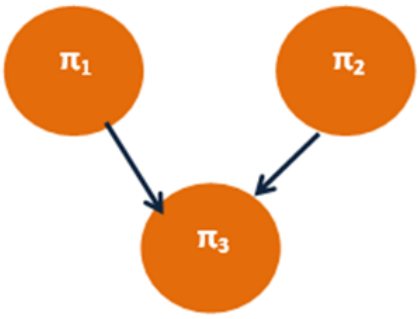

Figure 2.6: *A simple digraph with* 3 *nodes.*

On the other hand, we need to normalized it because the out-degree of the nodes has influence on their *PageRank*. For example, for the graph in Figure 2.7, we will have,

$$\pi_3 = \pi_1/3 + \pi_2/2. \tag{2.2}$$

The importance of nodes 1 and 2 are divided by the number of their out-going edges.

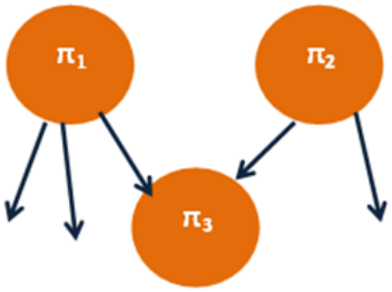

Figure 2.7: *A simple digraph with* 3 *nodes.*

Let us π^T be a vector and H be a matrix. If we have,

$$\overrightarrow{\pi}^T H = \lambda \overrightarrow{\pi}^T, \tag{2.3}$$

where λ is an eigenvalue of H, then nonzero vector π^T is the eigenvector of H.

We call the *PageRank* matrix H and we define it as,

$$H_{i,j} = \begin{cases} 1/o_i & \text{if } i \to j, \\ 0 & \text{if } i \nrightarrow j. \end{cases}$$

Here o_i is out-degree of node i.

2.1.6.2 Dangling Nodes

Consider the Figure 2.8. In this example, Node 4 has no out-going link to other nodes. These kind of nodes are called *dangling node* [Ipsen and Selee (2007)].

We will construct matrix H for this graph,

$$H = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1/3 & 0 & 1/3 & 1/3 \\ 1/3 & 1/3 & 0 & 1/3 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

We will have,

$o_4 = 0$,

$\pi^T = \pi^T H$,

$\pi_1 = \pi_3/3 + \pi_2/3$,

$\pi_2 = \pi_3/3$,

$\pi_3 = \pi_2/3$,

$\pi_4 = \pi_1 + \pi_3/3 + \pi_2/3$,

$\Rightarrow \pi_i = 0$.

The only way to have a solution is that all $\pi_i = 0$.

We want find $\pi_i \neq 0$ for every i, to achieve this goal we use so-called mandatory score-spreading. We modify the graph such that node 4 is pointed to all other nodes including itself. Now the fourth row of H will be

$$\begin{bmatrix} 1/4 & 1/4 & 1/4 & 1/4 \end{bmatrix}$$

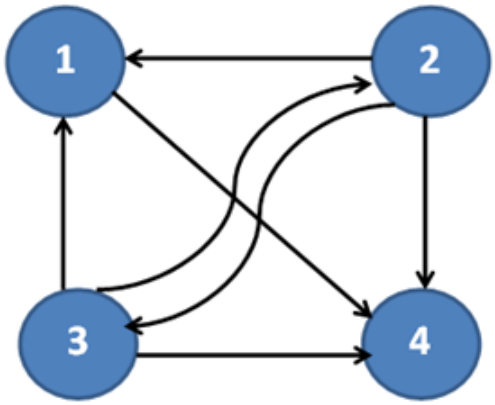

Figure 2.8: *A simple digraph with* 4 *nodes.*

or

$$1/4\begin{bmatrix}1 & 1 & 1 & 1\end{bmatrix}.$$

Matrix W;

$$W = \begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^T,$$

is the *indicator vector*[1] of a dangling node. The product of this matrix and fourth row of H gives us what we want:

$$1/4\begin{bmatrix}0\\0\\0\\1\end{bmatrix}\begin{bmatrix}1 & 1 & 1 & 1\end{bmatrix} = \begin{bmatrix}0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\0 & 0 & 0 & 0\\1/4 & 1/4 & 1/4 & 1/4\end{bmatrix}.$$

Using the above, we define modified matrix H, to be denoted by H',

$$H' = H + (1/N)W\overrightarrow{1}^T. \tag{2.4}$$

[1]If S is a finite set, $S = \{s_1, s_2, \cdots, s_n\}$, then the indicator vector of a subset T of set S is $x_T = (x_1, x_2, \cdots, x_n)$ where $x_i = 1$ if $s_i \in T$ and $x_i = 0$ if $s_i \notin T$.

We know that the biggest eigenvalue is 1, because the elements of each row of matrix H are zero or $1/o_i$ or $1/N$ (for dangling nodes), and for every node the sum of all $1/o_i$ or $1/N$ is equal to 1. This means H is a right stochastic matrix[1] and H and H^T have the same eigenvalues. So we will have,

$$\overrightarrow{\pi}^T H = \overrightarrow{\pi}^T. \tag{2.5}$$

2.1.6.3 Randomization

Consider now a surfer who is visiting in a particular web page. Assume that he chooses to follow one of the out-going links with probability θ, in which case he choose one of the out-going edges uniformly at random. With probability $(1-\theta)$ he actually may perform a random jumping to any other web page.

As an example, consider the graph in Figure 2.9.

To model this random jumping we write the following matrix,

$$\begin{bmatrix} 1/N & 1/N & 1/N & 1/N & \cdots \\ 1/N & 1/N & 1/N & 1/N & \cdots \\ 1/N & 1/N & 1/N & 1/N & \cdots \\ 1/N & 1/N & 1/N & 1/N & \cdots \end{bmatrix} = 1/N \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots \\ 1 & 1 & 1 & 1 & \cdots \\ 1 & 1 & 1 & 1 & \cdots \\ 1 & 1 & 1 & 1 & \cdots \end{bmatrix}.$$

This matrix is simply equal to $(1/N)\overrightarrow{1}\overrightarrow{1}^T$.

So, the matrix $(1-\theta)(1/N)\overrightarrow{1}\overrightarrow{1}^T$ models the random jumping.

[1] A right stochastic matrix is a real square matrix, with each row sums to 1.

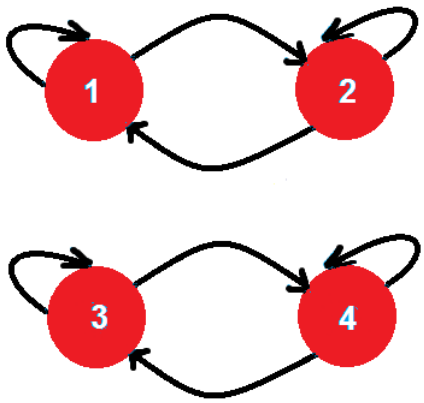

Figure 2.9: *An example of a surfer who is visiting a particular web page. If he starts from node* 1*, he may go to node* 2 *or jump to one of the nodes* 3 *or* 4

We now define matrix G, to be used in the *PageRank* algorithm, as

$$G = (1-\theta)(1/N)\,\overrightarrow{1}\,\overrightarrow{1}^{T} + \theta H'. \tag{2.6}$$

Here G is our new *PageRank* matrix.

2.1.6.4 *PageRank* Algorithm

Now we have matrix G,

$$G = (1-\theta)(1/N)\,\overrightarrow{1}\,\overrightarrow{1}^{T} + \theta(H + (1/N)W\,\overrightarrow{1}^{T}). \tag{2.7}$$

We want to find eigenvector $\overrightarrow{\pi}$ such that $\overrightarrow{\pi}^{T} = \overrightarrow{\pi}^{T}G$. For doing this we use Power Method [1].

We start with a nonzero initial π_i. From any initialization, after a

[1]The power method or power iteration is an eigenvalue algorithm which uses the given matrix G to produce a number λ (the eigenvalue) and a nonzero vector v (the eigenvector), such that $Gv = \lambda v$.[Buffalo (2015)]

number of iteration we will reach to the dominant eigenvalue [1] 1.

for $K = 0, 1, 2, \ldots,$

$\overrightarrow{\pi}^T[K+1] = (\overrightarrow{\pi}^T[K])G,$

$\vdots$

$\overrightarrow{\pi}^T = \overrightarrow{\pi}^T G.$

π is our PageRank vector,

$$\pi = \begin{bmatrix} a \\ b \\ c \\ \vdots \end{bmatrix}.$$

Here $a > b > c > \cdots$ [Asmussen (2003); Luxburg (2007)].

2.1.7 Dictionary Graph

A dictionary graph is a graph obtained from a dictionary. In this graph the vertices of the graph represent the head words of the dictionary and an edge is a link between two words if one word was used in the definition of the other word [Fukś and Krzeminski (2009)].

2.1.8 Correlation Coefficients

A correlation coefficient is defined as a measure which gives us a numerical value representing the degree of association between two variables. Furthermore, the correlation coefficient shows, how one variable changes with a

[1] If $\lambda_1, \lambda_2, \cdots, \lambda_n$ be the eigenvalues of an $n \times n$ matrix G and $|\lambda_1| > |\lambda_i|$ for $i = 2, \cdots, n$, λ_1 is called the dominant eigenvalue of G. The dominant eigenvectors of G are the eigenvectors corresponding to λ_1.

change of the value of the other variable.

This measure is a value which changes between -1 to $+1$. When it is positive, that means that there is an increasing relationship and when it has negative values that means that there is a decreasing relationship [Buxton (2008)].

If we have a series of n measurements of X and Y which we write as x_i and y_i for $i = 1, 2, \ldots, n$, then the sample correlation coefficient can be used to estimate the population Pearson correlation r between X and Y. The following formulas define the sample correlation coefficient [Wikipedia (2014a)].

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{(n-1)s_x s_y} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}. \quad (2.8)$$

In the above $\overline{x}$ and $\overline{y}$ are the sample means of x and y, and s_x and s_y, are the sample standard deviations of x and y. This can also be written as

$$r_{xy} = \frac{\sum_{i=1}^{n} x_i y_i - n\overline{xy}}{(n-1)s_x s_y} \quad (2.9)$$

or,

$$r_{xy} = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2}}. \quad (2.10)$$

2.1.9 Standard Deviation

Standard Deviation denoted by δ is a measure that we use to quantify the amount of variation of a set of data. A high standard deviation means that the data points tend to be far from the mean and they are spread out over a wider range of values.

Let us consider X as a variable which takes random values from a finite set $x_1, x_2, \ldots, x_N$, the standard deviation is,

$$\delta = \sqrt{1/N[(X_1 - \mu)^2 + (X_2 - \mu)^2 + \cdots + (X_N - \mu)^2]}, \tag{2.11}$$

where $\mu = 1/N(x_1 + x_2 + \cdots + x_N)$. Here we assume that each value has the same probability [Altman and Bland (2005); Croxton and Cowden (1956)].

2.1.10 Random Graph

One of the most important kinds of graphs is a random graph. Random graphs are especially important as models of web graphs. Random graphs have been extensively studied in modern graph theory and theoretical computer science.

Let $G(n, p)$ be a random graph with vertex set V, where p is the probability of connection between two distinct vertices, independently of other connections. Hence, the number of elements of V is a fixed number, but the number of edges can vary according to a binomial distribution with expectation $\binom{n}{2}p$. $G(n, p)$ is a random graph with n vertices and edge probability p [Aldous and Fill (2002)].

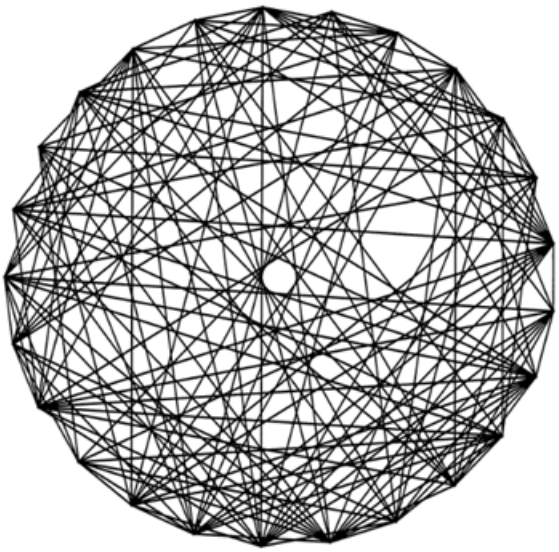

Figure 2.10: *A graph with 25 vertices, where edges are drawn with probability* 1/2.

2.1.11 Social Graph

The social graph is a type of graph that represents some kind of personal relations between internet users. By using of the word "graph" which has been taken from Graph Theory we emphasize that accurate mathematical analysis will be performed as opposed to the relational representation in a social network. The social graph has been referred to as "the global mapping of everybody and how they're related" [Wikipedia (2014c)].

2.1.12 Dynamic Graphs

Dynamic Graphs are Graphs which change by time. There are two kind of Dynamic Graphs:

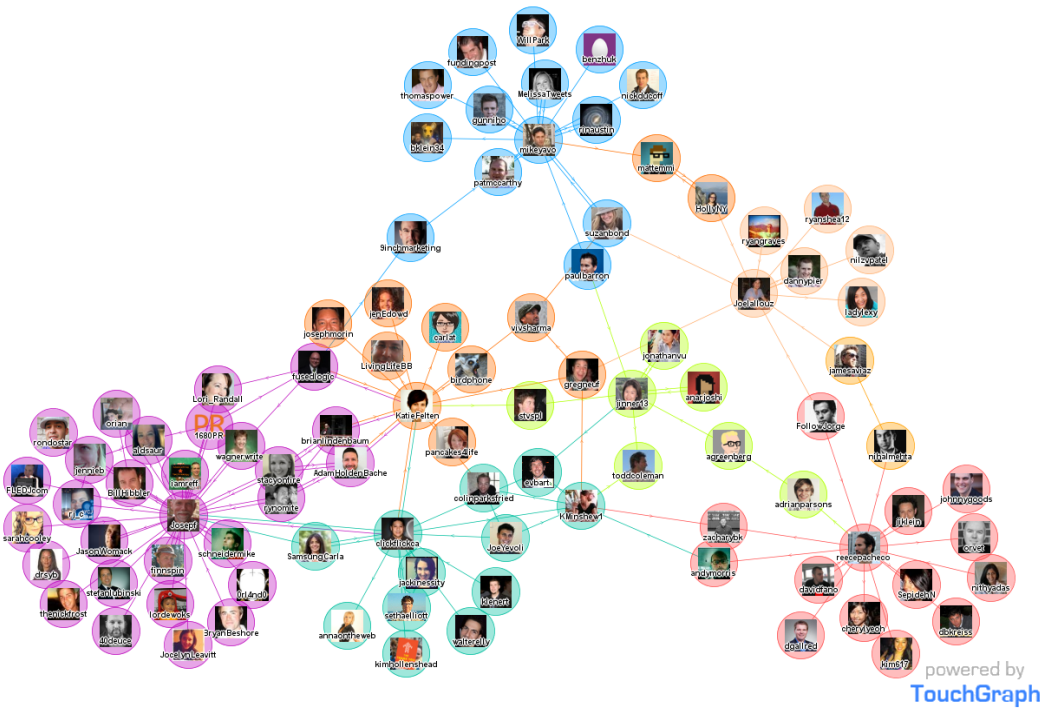

Figure 2.11: *Visualization of Twitter activity of data from popular social networking tool Hashable [touchgraph (2014)].*

1. Partially Dynamic Graphs: These graphs are subjected to just one kind of change, ie, either insertions or deletions, but not both of them.

2. Fully Dynamic Graphs: These graphs are subjected to a mix of insertions and deletions sequentially [Eppstein et al. (1992)].

2.1.13 *PageRank* in Dynamic Graphs

There are some algorithms which have been designed to include the changes of graphs in computing the *PageRank* but all of these algorithms depend on some basic knowledge of all the changes in the network as an assumption. These assumptions are not accurate in many modern applications of *PageRank*, and they are often not reasonable because they do not work in

real world situations, especially in Dynamic Graphs.

Algorithms used for computing *PageRank* can be divided in two categories: linear algebraic methods and Monte Carlo methods. Algorithms for updating *PageRank* changes have been proposed in both categories. We can name some of them, for example:

1. Random Probing: Random Probing is an algorithm which choses a random node uniformly at every step during the time. It is based on fact that at each point t, there is a most recent image of the graph, we call this image H^t (the meaning is that for every node v, the out-degree in H^t is the set of out-going edges observed when v was watched for the last time). Finally, we will have the *PageRank* vector of the mentioned image as our output. This algorithm is called Random Probing which is based on examining nodes with equal frequencies. The problem with this algorithm is that because the algorithm can probe a small part of the changing graph during each probe, it can not have an accurate up-to-date image of the graph.

2. Round-Robin Probing: Round-Robin Probing algorithm is another tool to take into account the changes in graphs. This algorithm moves through the nodes according a cycling order to probe the nodes. The output of this algorithm at any special time is the *PageRank* vector of the current image of the graph.

2.1.14 WordNet

2.1.14.1 About WordNet

Some of the graphs which we are going to analyze using *PageRank* are constructed form the WordNet database [WordNet (2014)]. WordNet is a huge English lexical database. It consists of nouns, verbs, adjectives and adverbs which are grouped into sets of cognitive synonyms (synsets).

Wordnet has about 117000 synsets which each of these sets expressing a distinct concept. There are links which connect Synsets by means of conceptual-semantic and lexical relations. The resulting network is a network of meaningfully related words and concepts which can be navigated with the browser.

Because of the structure of WordNet, it is a useful tool for computational linguistics and natural language processing. WordNet is similar to a thesaurus which categorizes words together according to their meanings. However, there are some important distinctions. First, WordNet interlinks not just word forms strings of letters but specific senses of words. Consequently, words that are found in close proximity to one another in the network are semantically disambiguated. Second, WordNet labels the semantic relations among words, whereas the grouping of the words in a thesaurus does not follow any explicit pattern other than meaning similarity.

2.1.14.2 WordNet Structure

Synonymy is the main relation which is defined between the words in WordNet. Synonym words that describe the same concept and are inter-

changeable in many contexts are categorized into unordered synsets. All of WordNets synsets are linked to other synsets by means of a small number of conceptual relations. Indeed, a synset contains a brief definition and, in most cases, one or more short sentences showing the use of the synset members. Word forms with several distinct meanings are represented in as many distinct synsets. It means that each form-meaning pair of words in WordNet is unique [WordNet (2014)].

2.2 Review of the previous related works

The main focus of this thesis is on *PageRank* algorithm and some of its properties in some different graphs. Although *PageRank* algorithm and its application have attracted a great amount of interest recently, it is a fairly new concept in graph theory. Although there is not a lot of work in this area, there are still some remarkable results. We just will mention here some of the works related to our research.

PageRank was introduced by Sergey Brin and Larry Page in their paper [Brin and Page (1998)]. According to Page and Brin, if a page received most links aimed to it that means this page is the most important page on the internet. In graph theory and mathematical context we use nodes instead of pages. Also because *PageRank* assigns probabilities to pages (nodes), the sum of all pages (nodes) *PageRanks* must be equal to one.

Brin and Page introduced the simplest formulation for *PageRank*. Suppose $PR(A)$ is the *PageRank* of page (node) A and A has pages (nodes) $T_1, \cdots, T_n$ which point to it. Then the *PageRank* of node A is,

$$PR(A) = (1-d) + d(PR(T_1)/C(T_1) + \cdots + PR(T_n)/C(T_n)), \quad (2.12)$$

where d is a damping factor[1] between 0 and 1.

Later in their next paper[Page et al. (1999)] they introduced a new description for *PageRank*. According to this new description *PageRank* is a method for ranking pages objectively and mechanically, and an effective tool to measure the human interest and attention devoted to pages. Also they made a comparison between *PageRank* and an idealized random web surfer and they showed a way to efficiently compute *PageRank* for number of pages. Furthermore, they introduced the application of *PageRank* in searching and in user navigation.

Pavel Berkhin has made a survey on *PageRank* computing [Berkhin (2005)]. He has reviewed a lot of research related to *PageRank* computing, for example algorithms shown in Figures 2.12, 2.13, 2.14.

Fortunato, Boguna, Flammini, and Menczer have tried to a good approximation for *PageRank* using some properties of the nodes. To achieve this goal they used in-degree of the pages [Fortunato et al. (2008)].

Instead of analyzing the *PageRank* of single pages, they used the category classes of pages. They defined these classes according to the degree of nodes, $k \equiv (k_{in}, k_{out})$ and they calculated the average *PageRank* of nodes for class k of nodes degree. They formulated the average *PageRank* as,

$$\overline{P}_n(k) = (1/N)P(k)\Sigma P_n(i), \quad (2.13)$$

[1]Damping factor is the probability at each page (node) the "random surfer" will get bored and jump to another random page.

Algorithm 1: Poor man's PageRank Algorithm

Input: Given a transition matrix P, a teleportation vector v, and a coefficient c
Output: Compute PageRank p
begin
  Initialize $p^{(0)} = v$, $k = 0$
  repeat
    $p^{(k+1)} = cP^T p^{(k)}$
    $\gamma = \|p^{(k)}\| - \|p^{(k+1)}\|$
    $p^{(k+1)} = p^{(k+1)} + \gamma v$
    $\delta = \|p^{(k+1)} - p^{(k)}\|$
    $k = k + 1$
  until $\delta < \epsilon$
end
return $p^{(k)}$

Figure 2.12: *Poor man's PageRank Algorithm[Berkhin (2005)].*

where $i \in k$ and k is the degree class. Class k means that all the nodes in this class have in-degree k_{in} and out-degree k_{out}. In their formula $P(k)$ is the probability that a node belongs to the degree class k. At the end, they concluded that *PageRank* has such a global nature that makes it very different from in-degree. So, we cannot calculate *PageRank* of the pages approximately while we do not know enough about the Web Graph globally. In addition, they showed that due to the weak degree-degree correlations in the Web link graph, the correlation between *PageRank* and in-degree is very strong meaning that these two measures give us very similar information, especially for the most popular pages.

Litvak, Scheinhardt, and Volkovic introduced a novel mathematical model to explain the relation between *PageRank* and in-degree [Litvak et al. (2009)]. They modeled this relation through a stochastic equation, which is based on the original definition of *PageRank*. They obtained the following formula,

Algorithm 2: Block Structure Method (*blockRank*)

Input: Given a graph W over nodes $G = \bigcup G_I$, $I = 1 : N$,
a teleportation vector v, and a coefficient c
Output: Compute PageRank p
begin
  for $I = 1 : N$ **do**
    let P_{II} be a transition matrix over block G_I
    for $i \in G_I$ set $v_{I,i} = v_i / \sum_{j \in G_I} v_j$
    $l_I = pageRank(P_{II}, v_I, v_I)$
  for $I = 1 : N$ **do**
    $\tilde{v}_I = \sum_{i \in G_I} v_i$
    for $J = 1 : N$ **do**
      $\tilde{L}_{IJ} = \sum_{i \in I, j \in J} P_{ij} l_i$
  let $\tilde{P}$ be a transition matrix corresponding to weighted block structure $\tilde{L}$
  $b = pageRank(\tilde{P}, \tilde{v}, \tilde{v})$
  set $s = (b_1 l_1, b_2 l_2, \ldots, b_N l_n)$
  $p = pageRank(P, s, v)$
end
return p

Figure 2.13: *Block Structure Method[Berkhin (2005)].*

$$R \stackrel{d}{=} c \sum_{J=1}^{M} (1/d) R_j + (1 - c), \tag{2.14}$$

where R is the *PageRank* of a page which has been chosen randomly, M is the in-degree of the chosen random page and c is the damping factor. Because they wanted to focus on the influence of in-degree, without considering other factors, they assumed the number of out-degree for all pages is equal to $d \geqslant 1$. Also they concluded that distribution of *PageRank* and in-degree should follow power laws with the same exponent.

Ghosh, Kuo, Hsu, Lin, and Lerman tried to consider the dynamic nature of networks and present a way to find the important nodes in an evolving graph [Ghosh et al. (2011)]. They wanted to do the time-aware ranking in dynamic citation networks, and they introduced two time-aware metrics methods to rank the publications in a citation network. They named the

Algorithm 3: OPIC

Input: Given link data L
Output: Compute PageRank p
begin
  Initialize $c_i = 1/n$, $h_i = 0$, $H = 0$
  repeat
    select i randomly with non-zero probability
    $h_i += c_i$
    for *each* $i \to j$ **do**
      $c_j += c_i/deg(i)$
    $H += c_i$
    $c_i = 0$
  until *converged*
  for *each* i **do**
    $p_i = (h_i + c_i)/(H + 1)$
end
return p

Figure 2.14: *OPIC Algorithm[Berkhin (2005)].*

methods Efficiently Computing Matrix ECM and Retained Adjacency Matrix RAM. Application of these methods is when more recent nodes are more important for example the published papers.

RAM is a method which gives a greater weight to a more recent node and degrades the weights as time pass. This means the weight of the nodes decreases as they ages.

ECM is a method to score the nodes at the end of a time period and rank them according to their scores. It measures the number of citation chains between nodes. In this method the chains are weakened not only by their length, but also by the age of the citing nodes. ECM In comparison with RAM it considers extra penalty for the chains length.

Ryan Rossi and David Gleich tried to modify the *PageRank* formulation for dynamic graphs [Rossi and Gleich (2012)]. Actually they wanted to study the dynamic *PageRank* using Evolving Teleportation. They introduced a new algorithm for this purpose.

Their algorithm intended to show how the *PageRank* (importance of a

page (node)) changes by external interest influence. It considered *PageRank* as a dynamical value and uses a teleportation vector to represent the changes. In their algorithm, P is defined as the transposed transition matrix for a random-walk on a graph, where

$P_{j,i}$ = probability of transitioning from node i to node j.

$$P = A^T D^{-1}, \tag{2.15}$$

where D is a diagonal matrix. The elements of D are the degrees of each node on the diagonal.

According to the classical standard *PageRank* algorithm,

$$x(k+1) = \alpha P x(k) + (1-\alpha)v. \tag{2.16}$$

α = damping parameter in *PageRank*. for any $0 \leq \alpha < 1$.

v= teleportation distribution vector.

For any teleportation distribution vector v, $v_i \geq 0$ and $\Sigma v_i = 1$.

x= The *PageRank* vector which is the solution to the *PageRank* computation.

Rearranging the above, they obtained

$$\Delta x(k) = x(k+1) - x(k) = \alpha P x(k) + (1-\alpha)v - x(k) = (1-\alpha)v - (I-\alpha P)x(k). \tag{2.17}$$

I= Identity matrix.

Thus, changes in the *PageRank* values at a node evolve depending on the value $(1-\alpha)v-(I-\alpha P)x(k)$. They reconsidered this update as a continuous time dynamical system,

$$x'(t)=(1-\alpha)v-(I-\alpha P)x(t). \tag{2.18}$$

Other iterative methods also give rise to related dynamical systems. In the dynamic teleportation model, v is no longer fixed, but is instead a function of time $v(t)$,

$$x'(t)=(1-\alpha)v(t)-(I-\alpha P)x(t). \tag{2.19}$$

This means the *PageRank* values $x(t)$ may not settle or converge.

Using standard texts on dynamical system, they had,

$$x(t)=exp[-(I-\alpha P)t]x(0)+(I-\alpha P)\int_0^t exp[-(I-\alpha P)(t-\tau)]v(\tau)d\tau. \tag{2.20}$$

If $v(t)=v$ is constant with respect to time, then

$$\int_0^t exp[-(I-\alpha P)(t-\tau)]v(\tau)d\tau=(I-\alpha P)^{-1}v-exp[-(I-\alpha P)t](I-\alpha P)^{-1}v. \tag{2.21}$$

For constant $v(t)$,

$x(t) = exp[-(I - \alpha P)t](x(0) - x) + x.$

v(t)= a teleportation distribution vector at time t.

x(t)= solution to the Dynamic *PageRank* computation for time t.

$x'(t)$= derivation of $x(t)$.

We know x is the solution to static *PageRank*, is given by

$(I - \alpha P)x = (1 - \alpha)v.$

All the eigenvalues of $-(I - \alpha P)$ are negative, and that causes the matrix exponential terms disappear in a sufficiently long time horizon. Thus, when $v(t) = v$, nothing has changed. They recovered the original *PageRank* vector that we named x as the steady-state solution.

Then they had,

lim $x(t) = x$, when $t \to \infty$.

Note: x is the *PageRank* vector.

This showed them that what they called a dynamic teleportation *PageRank* is a generalization of our original PageRank vector.

Other people who worked on *PageRank* in Dynamic Graphs were Mahdian, Bahmani, Kumar and Upfa. They summarized the result of their research in a paper focused on *PageRank* on evolving graph [Bahmani et al. (2012)].

They thought that although there are many algorithms which have been designed to discover the most important nodes in dynamic graphs, like various centrality metrics and *PageRank*, most of them did not take into account that the networks naturally are dynamic.

They believed that in real world the structure of many complex networks is dynamic and it supposed to change. According to them we expect that the networks nodes and edges will be created or vanished over the time. On

the other hand, *PageRank* depends on the structure of the graph and this structure is evolving in time.

To start the discussion about the *PageRank* in dynamic graphs, they first recalled the definition of *PageRank*.

Let

π = *PageRank* vector of graph G, and

π^t = The real *PageRank* vector of graph G at the end of the time step t.

When one considers computing *PageRank* in a dynamic graph, the main goal is to design an algorithm resulting in *PageRank* of a changing directed graph. They set Φ^t as the *PageRank* vector for the graph at the end of each time step, resulted by their algorithm. They wanted to compute that in such a way that the difference between π^t and Φ^t is small. The small difference between π^t and Φ^t means that the computed *PageRank* by their algorithm is not very different from the real *PageRank* vector of the graph G.

Mentioning some previous algorithms like Random Probing and Round-Robin Probing, they introduced two new algorithms for computing *PageRank* in dynamic graphs. Their believed that their algorithms uses better realistic assumption.

1. Proportional Probing: Proportional Probing (introduced by Mahdian, Bahmani, Kumar and Upfal): this algorithm is based on this fact that when the outgoing edges of nodes with high *PageRank* change, the *PageRank* of other nodes changes a lot.

   It starts probing the nodes at each step of the time t by choosing a node v with probability proportional to its *PageRank* in the algorithm's current image of the graph. The selection of nodes is according to the proportional to their *PageRank*, but in a stochastic manner.

2. Priority Probing: Priority Probing (introduced by Mahdian, Bahmani, Kumar and Upfal): This algorithm examines nodes with frequencies proportional to their current *PageRank*. It attempts to assign a priority to each node.

Today, *PageRank* is still a very easy to compute and useful method to categorize nodes in both web graphs and social networks. Although a lot of work has been done in the field of web information retrieval, search engines still use the *PageRank* algorithm for ranking search results.

Chapter 3

Modeling the Networks

3.1 Introduction

Although an extensive of research has been performed on the relation between in-degree of nodes and *PageRank*, we are not aware of any research about the correlation between total degree and the *PageRank*.

In this work we investigate the relationship between degrees of individual nodes and their *PageRank*. We also study some other properties of *PageRank* of the nodes, using several different kinds of graphs as examples. First we start with analyzing some selected dictionary graphs, finding *PageRank* of their vertices, and comparing it with degrees of vertices. We then compute the correlation between the *PageRank* and degrees..

The underlying hypothesis was that more important words are likely to receive more links from other words. More frequent words have higher degree and because of the nature and form of the dictionary graphs they will have higher *PageRank*. That means there must be a high correlation between *PageRank* and degree of every node. The results presented subsequently support this hypothesis.

Then we tried to check some properties of *PageRank* in dictionary graphs. For example, we checked that what will happen with the ranking of the nodes according to their *PageRank* if we randomly omit some words (nodes). The results did not show any special trends. Omitting random words does not cause significant change in the nodes ranking, but choosing special words with critical location in the graph can lead to interesting results. We found that if we omit some chosen nodes, some other words will receive very low rank in *PageRank* ranking. All of this emphasizes the critical role of the structure of the graph in computing the *PageRank*.

In the second phase, we started to analyze some different kinds of graphs, like social graphs and random graphs. We calculated the *PageRank* of the nodes for all of these graphs and the correlation between the degree and *PageRank*. We saw that the correlation between *PageRank* and degree was low. Actually, in the case of Preferential Attachment graphs the correlation was negative. Then we omitted some nodes and checked the results again. We got the same results, low correlation for random graphs and negative correlation for Preferential Attachment graphs.

Finally we compared the social graphs and random graphs and we saw that the random graphs were more resistant to the *PageRank* changes than the Preferential Attachment graphs.

3.2 Correlation between *PageRank* and Total Degree

Let us start with recalling some formulas. We know the matrix used in *PageRank* algorithm is given by

$$G = (1-\theta)(1/N)\overrightarrow{1}\overrightarrow{1}^T + \theta(H + (1/N)W\overrightarrow{1}^T),$$

and the correlation coefficient between two variables is

$r_{xy} = \frac{\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\overline{x})^2\sum_{i=1}^{n}(y_i-\overline{y})^2}}$.

Now, let D_i be the total degree of node i and P_i be its *PageRank*. Then we can define the correlation between them as

$$C_{PD} = \frac{\sum_{i=1}^{n}(P_i-\overline{P})(D_i-\overline{D})}{\sqrt{\sum_{i=1}^{n}(P_i-\overline{P})^2\sum_{i=1}^{n}(D_i-\overline{D})^2}}. \tag{3.1}$$

3.2.1 Dictionary Graphs

Merriam Graph is the graph of Merriam-Webster dictionary with 93062 nodes and 1237131 edges. It is well known that in a typical text, a small set of high-frequently words can cover a remarkable part of the text. It is obvious that in the case of highly frequent words we can assume the total degree of every node is equal to its in-degree, because in comparison to their in-degree, their out-degree is very small. So, high frequency words which have high in-degree (we took total degree instead of in-degree) will be the high ranked words according to their *PageRank*. That is because we define the *PageRank* of the words as their importance and in a Dictionary Graph more important words are more frequent words.

We used Gephi software to find the degree and *PageRank* of the nodes. Then we used SPSS to calculate the C_{PD}.

Analyzing the results we found a high correlation equal to 0.97 between *PageRank* and degree of the nodes' of Merriam-Webster dictionary. Also we did the same for a small dictionary with 45202 words (nodes) and 551940 edges. We got a low correlation between the nodes' degree and their

PageRank equal to 0.569. Repeating the mentioned steps for Dictionnaire de l'Académie francaise with 21265 nodes and 480234 edges resulted in $C_{PD} = 0.856$ which is still a high correlation.

Then we examined two different kinds of WordNet dictionaries. A Hypernym dictionary and a Synonym dictionary, respectively with 122653 nodes, 200211 edges and 153745 nodes, 337012 edges. Hypernym is the more general meaning of a word or may be it is better to be said the general category of a word. For example; the word beverage is a hypernym of tea.

After computing the C_{PD} for these dictionaries we found very low correlation between their nodes' *PageRank* and degree. Synonym dictionary's $C_{PD} = 0.691$ and Hypernym dictionary's $C_{PD} = 0.484$.

So as a result it can be said the structure and properties of dictionary graphs causes some changes in computing their nodes' *PageRank*. For example, we can assume the total degree of every node is equal to its in-degree. This will be resulted in a high correlation between total degree and *PageRank* of the nodes. Also, it seems that more well-defined a dictionary, higher C_{PD} it has.

3.2.2 Preferential Attachment and Random Graph

We continued our research by creating about 20 different graphs categorized in 2 distinct kinds of graphs (10 graphs of each kind): Preferential Attachment [Barabasi and Albert (1999)] which is a kind of social graph and Erdos-Renyi network which is a Random graph.

1. Preferential Attachment: In some networks, a few "hubs" have lots of

connections, while everybody else only has a few. This model shows one way such networks can arise.

We created our Preferential Attachment graphs with about 100 nodes and 100 edges. This means whenever we added a node to this graph, it just connected to one other node which has chosen randomly with one link. All of these mean the in-degree of all nodes was 1 and the out-degree of a lot of nodes was 0. The degree distribution of this graph showed that we had a few nodes with high degree (out-degree), some hubs, and most of the nodes had low degree.

2. Random Graph: In this graph, $(G(n,p))$, each possible link is given a fixed probability of being created.

   We created our Random graphs with about 100 nodes and edges with probability of 1/100 . This means every edge connected to the nodes with probability of 1/100. This produced a graph with 100 nodes and about 100 edges. Also we saw the in-degree or out-degree of some nodes was 0 while the degree distribution of this graph was more smooth and normal. We did not have any noticeable hub.

After that we calculated C_{PD} for both kinds of graphs and amazingly it was very low for all of them.

Especially in the case of Preferential Attachment graphs C_{PD} was even negative. As we know the *PageRank* has a high correlation with in-degree of the nodes and here in-degree of all nodes is equal to 1. So, the location of the node and its out-degree plays the main role.

In Random graphs it happens because of the probability of the edges. Actually if we have n nodes and edges with with $P = 1/n$, there is not

noticeable difference between the nodes' degree (neither their in-degree nor out-degree). Here the structure of the graph and the location of the node plays the main role in its *PageRank*.

3.3 *PageRank* Resistance Against the Change

Let us consider a graph G with n nodes and e edges. We define $n = \{A_1, A_2, \ldots, A_n\}$ and $PR_{A_1} = PageRank$ of node A_1 which has k incoming neighbors. Here d is our damping factor that is equal to 0.85, $N_1(A_1)$ =the first incoming neighbor of node A_1 and $O_{N_1(A_1)}$ is the out-degree of the first incoming neighbor of node A_1. Then, we will have,

$$PR_{(A_1)} = (1-d) + d(PR_{N_1(A_1)}/O_{N_1(A_1)} + PR_{N_2(A_1)}/O_{N_2(A_1)} + \cdots + PR_{N_k(A_1)}/O_{N_k(A_1)}),$$

$$PR_{(A_2)} = (1-d) + d(PR_{N_1(A_2)}/O_{N_1(A_2)} + PR_{N_2(A_2)}/O_{N_2(A_2)} + \cdots + PR_{N_k(A_2)}/O_{N_k(A_2)}),$$

$$\vdots$$

$$PR_{(A_n)} = (1-d) + d(PR_{N_1(A_n)}/O_{N_1(A_n)} + PR_{N_2(A_n)}/O_{N_2(A_n)} + \cdots + PR_{N_k(A_n)}/O_{N_k(A_n)}).$$

Let us omit randomly m nodes of this graph. Then we will have, $n = \{A_1, A_2, \ldots, A_{n-m}\}$. We introduce $OP = (m/n) * 100$. So, after omitting m nodes, node A_1 is still a part of G with $(1 - OP)$ probability and it is the same for all nodes of G. We will have,

$PR'_{(A_1)} = (1-OP)[(1-d)+d(PR_{N_1(A_1)}/O_{N_1(A_1)}+PR_{N_2(A_1)}/O_{N_2(A_1)}+\cdots+$
$PR_{N_{k'}(A_1)}/O_{N_{k'}(A_1)})]$,
$PR'_{(A_2)} = (1-OP)[(1-d)+d(PR_{N_1(A_2)}/O_{N_1(A_2)}+PR_{N_2(A_2)}/O_{N_2(A_2)}+\cdots+$
$PR_{N_{k'}(A_2)}/O_{N_{k'}(A_2)})]$,
$\vdots$
$PR'_{(A_{n-m})} = (1-OP)[(1-d)+d(PR_{N_1(A_{n-m})}/O_{N_1(A_{n-m})}+PR_{N_2(A_{n-m})}/$
$O_{N_2(A_{n-m})}+\cdots+PR_{N_{k'}(A_{n-m})}/O_{N_{k'}(A_{n-m})})]$.

Let us denote the difference between the *PageRanks* of node A_1 before and after omitting n nodes of graph G by Δ. Then,

$\Delta_{(A_1)} = PR_{(A_1)} - PR'_{(A_1)}$,
$\Delta_{(A_2)} = PR_{(A_2)} - PR'_{(A_2)}$,
$\vdots$
$\Delta_{(A_{n-m})} = PR_{(A_{n-m})} - PR'_{(A_{n-m})}$.

As we know the sum of all nodes' *PageRank* for a graph G is equal to 1. So, when we omit some nodes, their *PageRank* will be distributed over the rest of the nodes. It means the every node's *PageRank* will be changed after omitting some nodes and Δ is not a good measure to describe the *PageRank* changes in this case. To solve this issue let us use standard deviation of Δs.
$\delta_\Delta = \sqrt{(1/n)[(\Delta_{(A_1)}-\mu)^2+(\Delta_{(A_2)}-\mu)^2+\cdots+(\Delta_{(A_n)}-\mu)^2]}$,
where $\mu = (1/n)(\Delta_{(A_1)}+\Delta_{(A_2)}+...+\Delta_{A_{(n-m)}})$.

3.3.1 Dictionary Graphs

We tried to find the nodes' *PageRank* changes after omitting some random nodes in our dictionary graphs. To achieve this goal we omitted about 100 nodes of every graph (we will omit 5% of nodes in the next graphs but in case of dictionary graphs it is impossible because of the large number of

nodes). Results showed us that if we rank the nodes according to their *PageRank*, the nodes in the bottom of the list will face more changes in comparison to the nodes on the top. Actually, the ranking and *PageRank* value of high-ranked nodes did not change significantly. It was expected because of the large number of these graphs nodes. On the other hand, by chosing some special high-ranked nodes which are in critical location in the graph it is possible to make huge changes in the nodes' *PageRank*.

3.3.2 Preferential Attachment and Random Graph

As we know in this kind of Preferential Attachment graph that we chose, the number of edges and nodes are equal and every node has just 1 incoming edge and a lot of nodes have no out-going links. Also, we have a few hubs with a high total degree while most of the nodes have the same degree.

So, let us name our Preferential Attachment graph as G with n nodes and n edges. We define the node set as $\{A_1, A_2, \ldots, A_n\}$ and $PR_{A_1} = PageRank$ of node A_1 which has 1 incoming neighbors. Here d is our damping factor that is equal to 0.85, $N(A_1)$ =the incoming neighbor of node A_1 and $O_{N_1(A_1)}$ is the out-degree of the incoming neighbor of node A_1. Then, we will have,

$PR_{(A_1)} = (1-d) + d(PR_{N(A_1)}/O_{N(A_1)})$,

$\vdots$

$PR_{(A_n)} = (1-d) + d(PR_{N(A_n)}/O_{N(A_n)})$.

If we omit randomly m nodes of this graph, we will have, *node set* $= \{A_1, A_2, \ldots, A_{n-m}\}$ and $OP = (m/n) * 100$. So, after omitting m nodes and repeating the previous computing, we will have,

$PR'_{(A_1)} = (1 - OP)[(1 - d) + d(PR_{N(A_1)}/O_{N(A_1)})]$,

$\vdots$

$PR'_{(A_{n-m})} = (1 - OP)[(1 - d) + d(PR_{N(A_{n-m})}/O_{N(A_{n-m})})]$,

$\Delta_{(A_1)} = PR_{(A_1)} - PR'_{(A_1)}, \cdots, \Delta_{(A_{n-m})} = PR_{(A_{n-m})} - PR'_{(A_{n-m})}$,

and finally,

$\delta_\Delta = \sqrt{(1/n)[(\Delta_{(A_1)} - \mu)^2 + (\Delta_{(A_2)} - \mu)^2 + \cdots + (\Delta_{(A_n)} - \mu)^2]}$,

where $\mu = (1/n)(\Delta_{(A_1)} + \Delta_{(A_2)} + \cdots + \Delta_{A_{(n-m)}})$

In case of the kind of Random graph that we have chosen, we have a graph G with n nodes and edges with probability $P = (1/n)$. The degree distribution of this graph is smooth and we do not have any hub.

$PR_{(A_1)} = (1 - d) + (d/n)(PR_{N_1(A_1)}/O_{N_1(A_1)} + \cdots + PR_{N_k(A_1)}/O_{N_k(A_1)}), \cdots,$

$PR_{(A_n)} = (1 - d) + (d/n)(PR_{N_1(A_n)}/O_{N_1(A_n)} + \cdots + PR_{N_k(A_n)}/O_{N_k(A_n)})$.

Omitting randomly m nodes, we repeat the computing, while *node set* $= \{A_1, A_2, \cdots, A_{n-m}\}$ and $OP = (m/n) * 100$. So, now we will have,

$PR'_{(A_1)} = (1-OP)[(1-d)+(d/n)(PR_{N_1(A_1)}/O_{N_1(A_1)}+\cdots+PR_{N_k(A_1)}/O_{N_k(A_1)})]$,

$PR'_{(A_{n-m})} = (1 - OP)[(1 - d) + (d/n)(PR_{N_1(A_{n-m})}/O_{N_1(A_{n-m})} + \cdots +$

$PR_{N_k(A_{n-m})}/O_{N_k(A_{n-m})})]$.

The formula for computing Δ and δ is the same as Preferential Attachment graphs.

In this step, we omitted 5 random nodes ($5\% * 100$ nodes) in all 20 created graphs and we computed the *PageRank* of the nodes. Then, we calculated the correlation between every node's *PageRank* before and after omitting the 5 nodes. This experiment resulted in a high correlation which means there was not any remarkable *PageRank* changes. Also, we compare the mean of changes for all 10 Preferential Attachment graphs and 10 Random graphs separately and we compared them together. It seemed the

PageRanks of Random graphs' nodes were more resistant to changes.

Finally it can be said although we have done a lot in this research, it is not entirely conclusive. If we want to reach a more reliable conclusion, we need to use bigger graphs as well as more types of them.

Our most important result is the critical role of the structure of a graph in computing the *PageRank* of the nodes. The changes of a node *PageRank* depend on a lot of factors. So just considering some properties of a node and examining them separately can not help us to find a good answer for our questions. *PageRank* is a property more related to a combined set of factors which must be considered together.

Chapter 4

Detailed Description of Experimental Results

4.1 Analyzing Dictionary Graphs

Merriam graph is the graph of Merriam-Webster dictionary with 93062 nodes and 1237131 edges. We used Gephi software to find the degree and *PageRank* of the nodes. Then we used SPSS to calculate the C_{PD}. As it is obvious from Figure 4.1, there is a high correlation equal to 0.97 between *PageRank* and degree of nodes of Merriam-Webster dictionary.

As it was expected the most frequent words had the highest *PageRank*. The most frequent words with highest *PageRank* were prefixes, suffixes, pronouns and auxiliary verbs. We omitted these words and then compared the words's *PageRank*. The results were the same, the most frequent words had the highest *PageRank*.

Then we computed the correlation for the first 12000 and 1000 words. Correlation was respectively 0.97 and 0.974, still high and almost the same

Correlations

		Degree	PageRank
Degree	Pearson Correlation	1	.970**
	Sig. (2-tailed)		.000
	N	93062	93062
PageRank	Pearson Correlation	.970**	1
	Sig. (2-tailed)	.000	
	N	93062	93062

**. Correlation is significant at the 0.01 level (2-tailed).

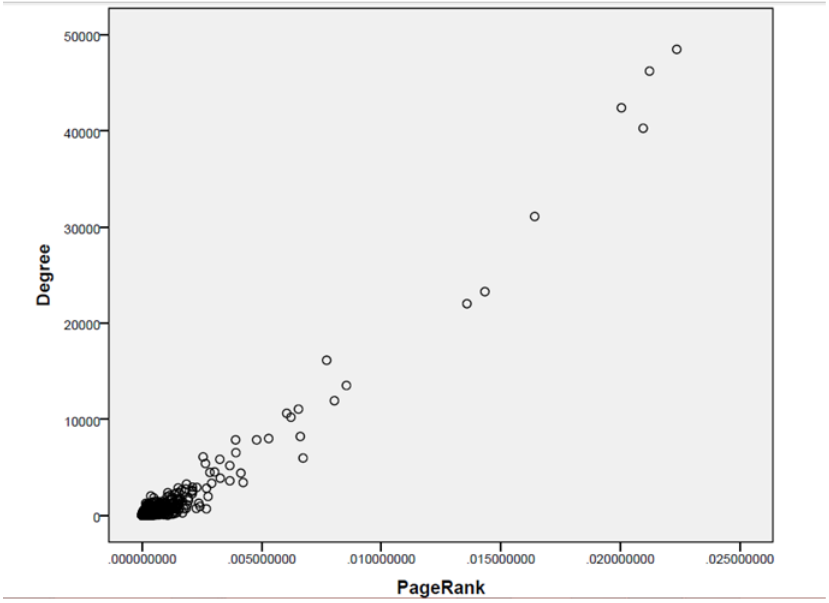

Figure 4.1: *Correlation between PageRank and degree of Merriam-Webster dictionary words.*

as the whole graph.

In next step we analyzed a French dictionary graph with 21291 nodes and 480234 edges. In comparison to Merriam-Webster dictionary it is a small dictionary but still there is a high C_{PR} for it (see Figure 4.2).

We did the same for the WordNet dictionary which is smaller than Merriam-Webster dictionary. It is a dictionary with 45204 words as nodes and 551941 edges. Amazingly the correlation between the *PageRank* and degree of nodes was low(see Figure 4.3). It can be said that this dictionary is more concise in definitions than Merriam-Webster dictionary and the

Correlations

		Degree	PageRank
Degree	Pearson Correlation	1	.856**
	Sig. (2-tailed)		.000
	N	21265	21265
PageRank	Pearson Correlation	.856**	1
	Sig. (2-tailed)	.000	
	N	21265	21265

**. Correlation is significant at the 0.01 level (2-tailed).

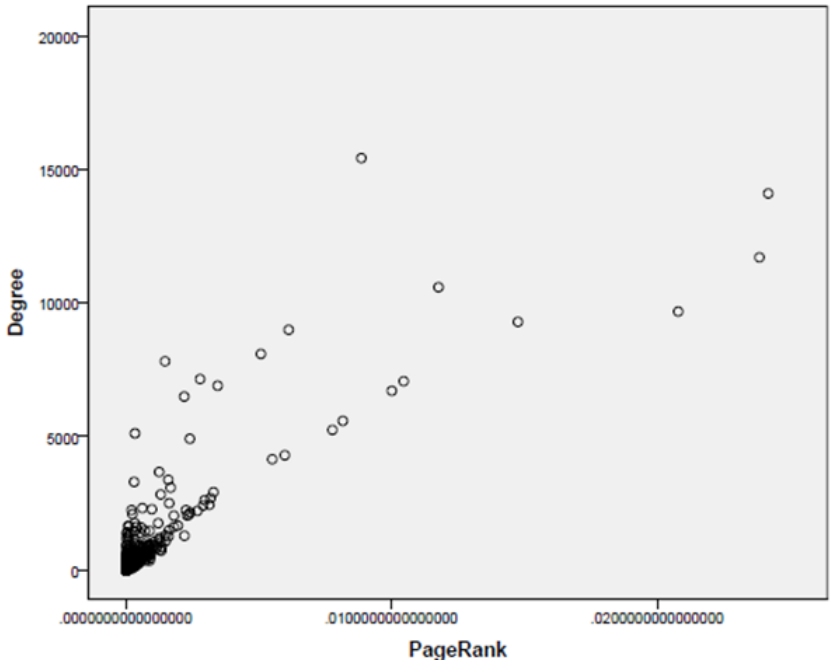

Figure 4.2: *Correlation between PageRank and degree of a French dictionary words.*

French one.

The next graphs that we analyzed were Hypernym with 29481 nodes and 47891 edges. Hypernym is a word or phrase whose semantic field is included within that of another word. According to definition of hypernym it is not a good example for a dictionary graph. the results proved it. Actually this dictionary was not well defined dictionary (see Figure 4.4).

Another graph that we studied was the WordNet synonyms dictionary

Correlations

		Degree	PageRank
Degree	Pearson Correlation	1	.569**
	Sig. (2-tailed)		.000
	N	45202	45202
PageRank	Pearson Correlation	.569**	1
	Sig. (2-tailed)	.000	
	N	45202	45202

**. Correlation is significant at the 0.01 level (2-tailed).

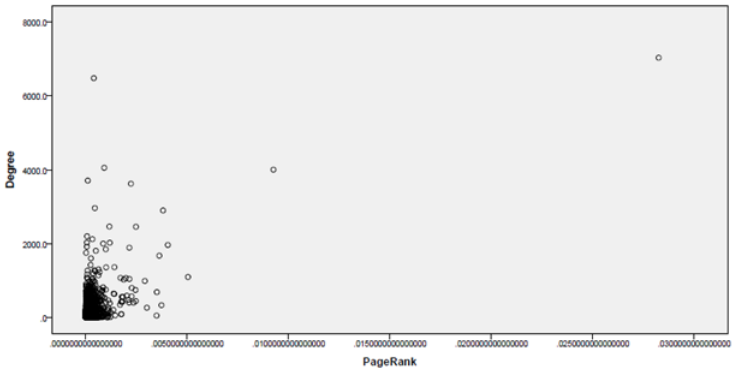

Figure 4.3: *Correlation between PageRank and degree of WordNet dictionary words.*

with 31249 nodes and 49651 edges. The structure of this graph is different from dictionary graphs and again it is not a good example for comparing the graphs (see Figure 4.5).

As a result of the this research it can be said that in dictionary graphs in addition to in-degree, there is a high correlation between the total degree of the nodes and their *PageRank*. Also, it can be said that if a dictionary has defined well, the correlation between the *PageRank* of the nodes and their degree must be high. Examining this hypothesis by testing it for more dictionaries should be a subject of further investigations.

Another issue which we studied was comparing the Merriam-Webster dictionary and French dictionary to check if the core high ranked words are

Correlations

		Degree	PageRank
Degree	Pearson Correlation	1	.484**
	Sig. (2-tailed)		.000
	N	122653	122653
PageRank	Pearson Correlation	.484**	1
	Sig. (2-tailed)	.000	
	N	122653	122653

**. Correlation is significant at the 0.01 level (2-tailed).

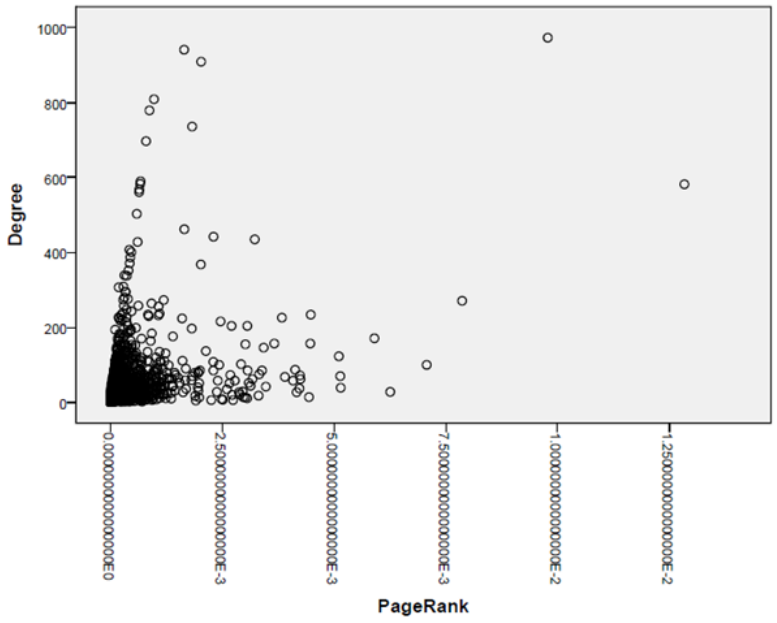

Figure 4.4: *Correlation between PageRank and degree of Hypernym dictionary words.*

the same in both dictionaries or not. Reaching a reliable conclusion, because of the big difference between the size of the dictionaries and grammatical difference between the structure of two languages, was not possible at this stage. Actually the comparison did not produce us any clear result (see Figure 4.6).

Also, we checked the nodes' *PageRank* changes after omitting some random nodes in our dictionary graphs. We omitted about 100 nodes of every graphs. Results showed us that if we rank the nodes according to

Correlations

		Degree	PageRank
Degree	Pearson Correlation	1	.691**
	Sig. (2-tailed)		.000
	N	153745	153745
PageRank	Pearson Correlation	.691**	1
	Sig. (2-tailed)	.000	
	N	153745	153745

**. Correlation is significant at the 0.01 level (2-tailed).

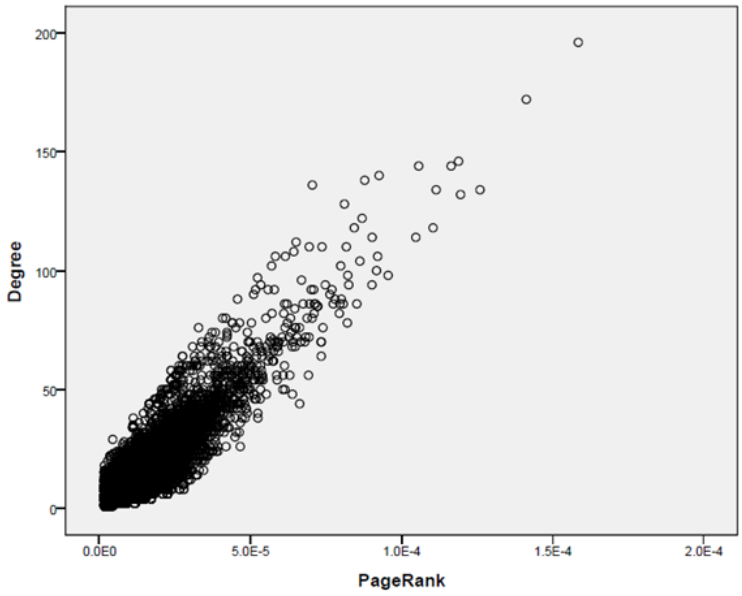

Figure 4.5: *Correlation between PageRank and degree of WordNet Synonyms dictionary words.*

their *PageRank*, the ranking and *PageRank* value of high-ranked nodes did not change significantly while the nodes in the bottom of the list changed more in comparison to the nodes on the top. Actually, It was expected because of the large number of these graphs nodes. On the other hand, if we want to make huge changes in the nodes' *PageRank*, we have to choose some special high-ranked nodes which are in critical location in the graph.

4.2 Analyzing some sample networks

4.2.1 Social Network

We created 10 different Preferential Attachment graphs with 100 nodes and 100 edges. The in-degree of all nodes was 1 and the out-degree of a lot of nodes was 0. The degree distribution of this graph showed that we had a few nodes with high degree (out-degree), some hubs, and most of the nodes had low degree (see Figure 4.7).

We calculated C_{PD} for all of them. C_{PD} value for all of them was negative. Figure 4.8 is a sample chart related to one of the examined graphs.

We continued our research with omitting 5% of nodes of the graphs and computed the *PageRank* again. Then we computed the correlation between every node's *PageRank* before and after omitting the 5 nodes. The high correlation showed us there was not any remarkable changes in *PageRank* values (see Figure 4.9).

4.2.2 Random Graph

We used Netlogo to form 10 different random graphs with 100 nodes and edges with probability of 1/100. In these graphs the in-degree or out-degree of some nodes was 0 while the degree distribution of this graph was smooth and normal. We did not have any noticeable hub (see Figure 4.10).

We calculated C_{PD} for all of them. C_{PD} value for these graphs was positive but less than 0.5 which is a low correlation (see Figure 4.11).

As we did for Preferential Attachment graphs, we omitted 5% of our

Random graphs nodes and we did the previous calculations again. We got high correlation and no significant change in the nodes' PageRank (see Figure 4.12).

At the end we compared the *PageRank* changes of Random graphs and Preferential Attachment graphs. The results showed us that the *PageRank* values in Random graphs generally are a little more resistant against the changes in comparison to Preferential Attachment graphs while they had a wider range of variability (see Figures 4.13,4.14,4.15).

In addition to the results presented so far, we studied the properties of some other kinds of graphs and their correlation with *PageRank*, but because we have not obtained any conclusive result we omitted them from the main chapters of the thesis.

For example:

1. We examined some other kind of graphs like a Social Network for Journalists using the Twitter API, a Preferential Attachment graph with 510 nodes and 508 edges and a Random Network with 500 nodes and 18814 edges which were created by probability 0.15. We calculated the correlation between *PageRank* and degree of these graphs' nodes. Also we tried to figure out what is going to happen for the *PageRank* of the nodes if we randomly omit some nodes.

   We found that the high ranked nodes have not changed remarkably. Actually in the case of all these graphs no remarkable changes occurred according to *PageRank* after omitting 5% of the nodes. The structure and formation of the Preferential Attachment graph and Random graph caused respectively a negative and a low positive correlation between *PageRank* and degree. [Appendix A]

2. We created 100 different graphs consists of 5 nodes. The graphs were directed but non weighted. In the first step we calculated the *PageRank* of all the nodes in different graphs in addition to other properties of nodes, In-Degree, out-degree, total degree and in-degree/out-degree.

   In the second step we omitted one of the nodes, node 3 and we calculated all of the mentioned measures, again. Then we introduced some new measures:
   Δ = the difference between the *PageRank* of a node before and after omitting the node 3.
   δ = Standard deviation of *PageRank* changes for every node.

   The goal was finding which nodes are more resistible against the change according to their *PageRank*. We compared the Δ and δ with in-degree, out-degree, total degree and in-degree/out-degree of the nodes.

   The result of researching theses graphs showed us that there is no special relation between these properties of nodes and their *PageRank* changes.

   In third step, we went further. We calculated the mean of all measures for the neighbors of all nodes for all 100 different graphs. Again we compared Δ and δ with new measures and we draw the charts showing the correlations.

   As a result it can be said that at least with examining a small graph like the graph which we chose, there is no obvious relation between the properties of nodes and delta. The only visible trend is the fact that the higher mean of a node's neighbors' *PageRank* causes a higher

change in its *PageRank* after omitting a node. [Appendix B]

No.	French words	Meaning	English words
1	quelque	some	see
2	sens	meaning	state
3	peu	little	act
4	fait	fact	person
5	homme	man	manner
6	parlant	speaking	word
7	sorte	kind	thing
8	nom	name	time
9	plusieurs	several	like
10	petit	little	same
11	mettre	put	part
12	partie	part	place
13	terre	earth	number
14	temps	time	hence
15	lieu	place	anything
16	mot	word	meaning
17	vieux	old	sometimes
18	marine	navy	action
19	grand	great	degree
20	plante	plant	quality
21	pronom	pronoun	something
22	personnel	staff	kind
23	substantivement	substantively	possession
24	mal	evil	make
25	porte	gate	limit
26	ouvrage	work	account
27	sous	under	especially
28	substantif	substantive	goods
29	parler	speak	object
30	familier	familiar	verb
31	maison	home	certain
32	rapport	report	made
33	toujours	always	cause
34	prendre	take up	sense
35	surtout	mainly	good
36	fort	strong	character
37	rendre	make	relation
38	usage	use	pronoun
39	pays	country	man
40	pierre	stone	condition
41	rien	nothing	small
42	vie	life	purpose
43	manger	eat	individualize
44	jurisprudence	jurisprudence	subject
45	deux	two	body
46	style	style	quantity
47	mauvais	bad	case
48	place	place	general
49	main	hand	singular
50	pluriel	plural	equivalent

Figure 4.6: *The comparisons between first* 50 *high-ranked words of French and English dictionaries according to their PageRank.*

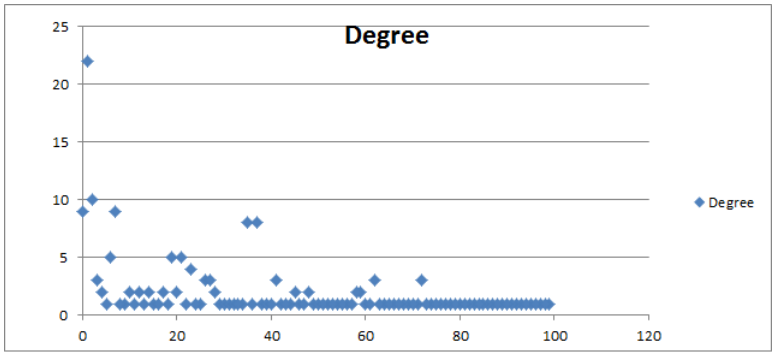

Figure 4.7: *Degree distribution of a Preferential Attachment graph.*

Correlations

		Degree	PageRank
Degree	Pearson Correlation	1	-.044
	Sig. (2-tailed)		.665
	N	100	100
PageRank	Pearson Correlation	-.044	1
	Sig. (2-tailed)	.665	
	N	100	100

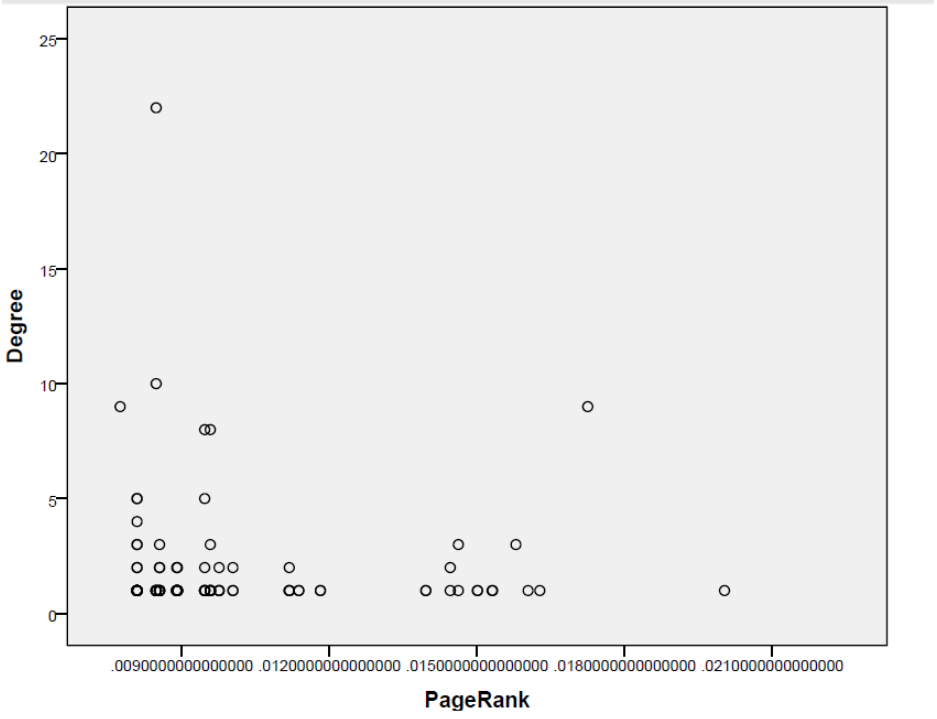

Figure 4.8: *Correlation between PageRank and degree of a Preferential Attachment graph.*

Correlations

		PageRankoriginal	PageRankafteromitting5ofnodes
PageRankoriginal	Pearson Correlation	1	.924**
	Sig. (2-tailed)		.000
	N	100	95
PageRankafteromitting5ofnodes	Pearson Correlation	.924**	1
	Sig. (2-tailed)	.000	
	N	95	95

**. Correlation is significant at the 0.01 level (2-tailed).

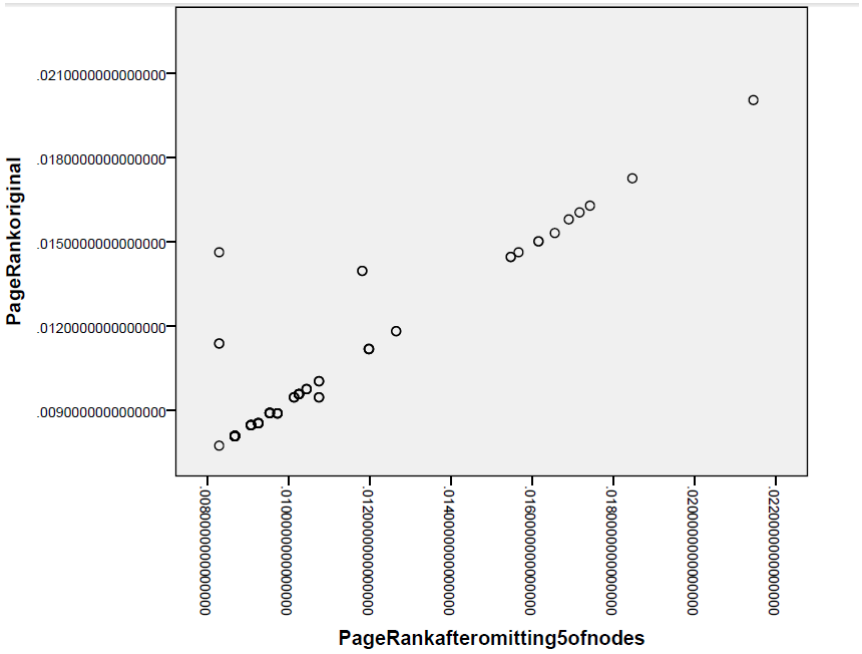

Figure 4.9: *Correlation between the nodes' PageRank of a Preferential Attachment graph.*

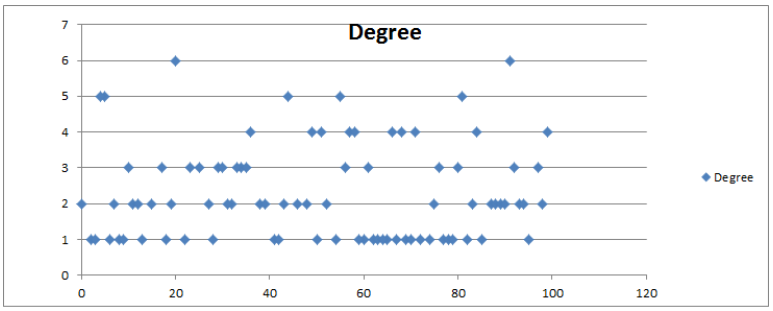

Figure 4.10: *Degree distribution of a Random graph.*

Correlations

		Degree	PageRank
Degree	Pearson Correlation	1	.295**
	Sig. (2-tailed)		.006
	N	86	86
PageRank	Pearson Correlation	.295**	1
	Sig. (2-tailed)	.006	
	N	86	86

**. Correlation is significant at the 0.01 level (2-tailed).

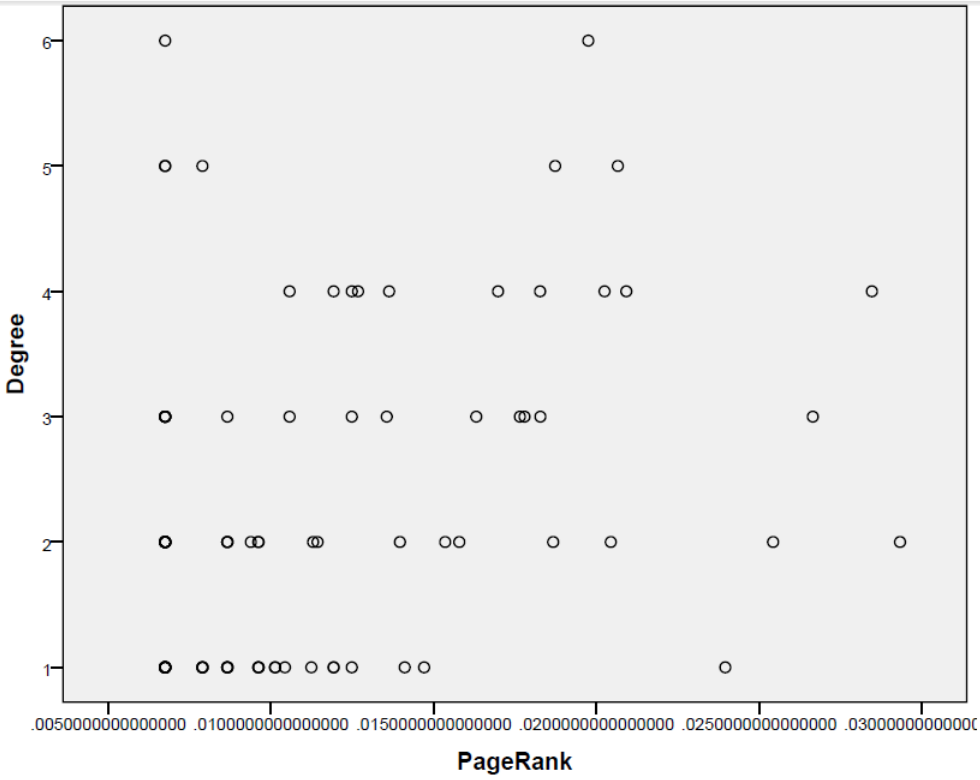

Figure 4.11: *Correlation between PageRank and degree of a Random graph.*

Correlations

		PageRankO	PageRankAO5
PageRankO	Pearson Correlation	1	.982**
	Sig. (2-tailed)		.000
	N	86	81
PageRankAO5	Pearson Correlation	.982**	1
	Sig. (2-tailed)	.000	
	N	81	81

**. Correlation is significant at the 0.01 level (2-tailed).

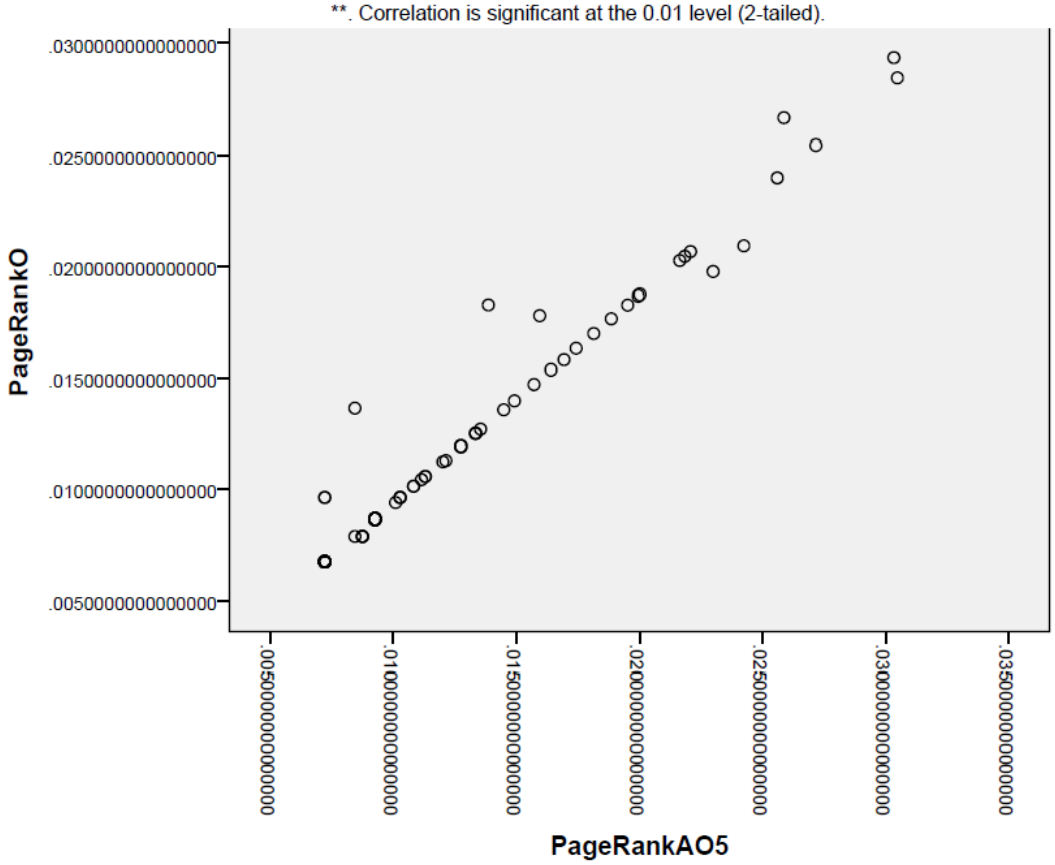

Figure 4.12: *Correlation between the nodes' PageRank of a Random graph.*

Graph No.	Prefrential Attachment	Random Graph
1	0.92	0.98
2	0.92	0.89
3	0.95	0.92
4	0.95	0.96
5	0.95	0.95
6	0.98	0.92
7	0.95	0.98
8	0.91	0.98
9	0.94	0.96
10	0.92	0.88
Mean	0.939	0.942

Figure 4.13: *Correlation between the nodes' PageRank changes of a Random graph and a Preferential Attachment graph.*

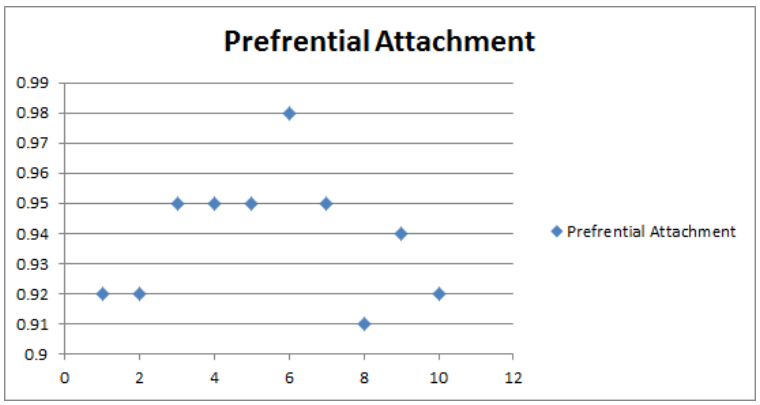

Figure 4.14: *Correlation distribution of nodes' PageRank changes for a Preferential Attachment graph.*

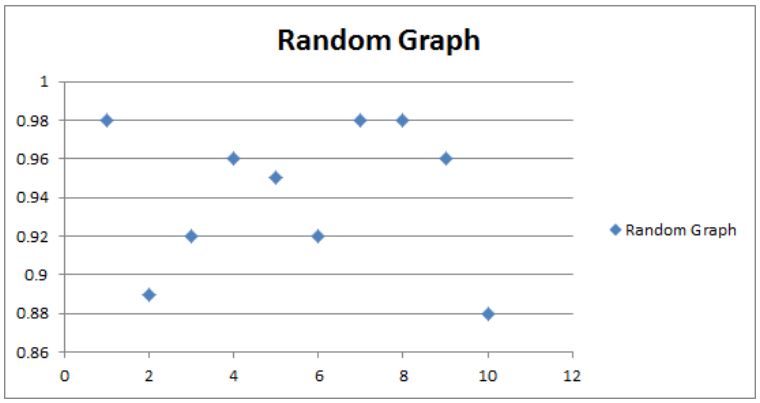

Figure 4.15: *Correlation distribution of nodes' PageRank changes for a Random graph.*

Chapter 5

Conclusions and Further Research

Due to the structural properties of dictionary graphs, vertices have very small out-degree comparing to their in-degrees. This results in a high correlation between the *PageRank* and total degree of nodes. So the structure of the graph plays a critical role in computing *PageRank*.

On the other hand, it seems that *PageRank* is a suitable tool to assess the quality of data which can be converted to a graph, for example, dictionaries. It seems that the quality of a dictionary is high, when the correlation between the *PageRank* of the nodes (words) and their degree is high, but a definitive conclusion in this matter clearly needs more research.

Another problem which we studied was comparing the Merriam-Webster dictionary and French dictionary to check if the core high ranked words are the same in both dictionaries or not. Drawing a reliable conclusion, because of the big difference between the size of the dictionaries and grammatical differences between the two languages, was not possible at this stage.

The hypothesis of a high correlation between the nodes' *PageRank* and their in-degree turned out to be wrong. In some kinds of graphs (like the Preferential Attachment graph), the aforementioned correlation does not

work.

In addition, the correlation between Node's *PageRank* and its degree for Preferential Attachment graphs was even negative and it was low in the case of the Random graph with the special properties that we defined.This emphasizes the critical role of the graph structure and nodes' location in their *PageRank*.

Furthermore, we saw that among all the properties of a node and its neighbors, just the *PageRank* of the neighbors of a node has a good correlation with its *PageRank* changes.

We also showed that omitting random nodes does not cause any special change in *PageRank* hierarchy of nodes. For achieving some goals like eliminating the importance of some nodes in graphs we have to choose and cut the nodes wisely. We can decrease the *PageRank* of a lot of nodes considerably just by deleting a few nodes.

For further research, the following investigations would be useful:

1. Checking the results for more dictionary graphs and comparing the quality of dictionaries by using the *PageRank* factor.

2. Examining different kinds and sizes of graphs and checking the changes after omitting random or selected nodes.

3. Studying the impact of other factors on a node's *PageRank*.

4. Studying the impact of a node's neighbors properties on its *PageRank* by using other kinds of Graphs.

5. Finding other types of graphs or networks whose quality could be determined by using *PageRank*.

6. Comparing the resistance of different graphs against changes according to their nodes' *PageRank*.

7. Studying effects of other kinds of changes (besides omitting some nodes) on the *PageRank* values.

Bibliography

Aldous, D. and Fill, J. A. (2002). Reversible markov chains and random walks on graphs. Unfinished monograph, recompiled 2014.

Altman, D. G. and Bland, J. M. (2005). Standard deviations and standard errors. *BMJ*, 331(7521):903.

Asmussen, S. (2003). *Applied Probability and Queues (Stochastic Modelling and Applied Probability)*. Springer.

Bahmani, B., Kumar, R., Mahdian, M., and Upfal, E. (2012). Pagerank on an evolving graph. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '12, pages 24–32, New York, NY, USA. ACM.

Bang-Jensen, J. and Gutin, G. (2002). *Digraphs: Theory, Algorithms and Applications*, volume 754. Springer Science and Business Media.

Barabasi, A. and Albert, R. (1999). Emergence of scaling in random networks. 11.

Berkhin, P. (2005). A survey on pagerank computing. In *Internet Mathematics*.

Bonato, A. (2008). *A Course on the Web Graph*, volume 184. American Mathematical Society, U.S.A.

Bondy, J. A. and Murty, U. S. R. (1976). *Graph Theory With Applications*, volume 381. Macmillan.

Brin, S. and Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. In *Seventh International World-Wide Web Conference (WWW 1998)*.

Buffalo, U. (2015). Power method— univirsity of buffalo, web page. [Online; accessed 2015].

Buxton, R. (2008). Statistics: Correlation. 8.

Croxton, F. E. and Cowden, D. J. (1956). *Applied general statistics*, volume 754. Sir Isaac Pitman and Sons.

Diestel, R. (2010). *Graph Theory*, volume 451. Springer-Verlag, Heidelberg.

Eppstein, D., Galil, Z., Italiano, G. F., and Nissenzweig, A. (1992). Sparsification — a technique for speeding up dynamic graph algorithms. In *Proc. 33rd Symp. Foundations of Computer Science*, pages 60–69. IEEE.

Fortunato, S., ná, M. B., Flammini, A., and Menczer, F. (2008). Algorithms and models for the web-graph. chapter Approximating PageRank from In-Degree, pages 59–71. Springer-Verlag, Berlin, Heidelberg.

Fukś, H. and Krzeminski, M. (2009). Topological structure of dictionary graphs. *Journal of Physics A: Mathematical and Theoretical*, 42:375101.

Galleryhip (2015). weighted-directed-graph— galleryhip, webpage. [Online; accessed 2015].

Ghosh, R., ting Kuo, T., nan Hsu, C., de Lin, S., and Lerman, K. (2011). Time-aware ranking in dynamic citation networks.

Gutenberg, P. (1996). Project gutenberg. the gutenberg websters unabridged dictionary. [Plainfield, N.J].

Harris, J., Hirst, J. L., and Mossinghoff, M. (2008). *Combinatorics and Graph Theory / Edition 2*, volume 381. Springer-Verlag New York.

Institute, T. B. (2015). web graph — the bordalier institute, webpage. [Online; accessed 2015].

Ipsen, I. and Selee, T. (2007). Pagerank computation, with special attention to dangling nodes.

Ipsen, I. and Wills, R. M. (2005). Analysis and computation of googles pagerank. *"7th IMACS International Symposium on Iterative Methods in Scientific Computing*, 35.

Litvak, N., Scheinhardt, W. R. W., and Volkovich, Y. (2009). In-degree and pagerank: Why do they follow similar power laws.

Luxburg, U. V. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17:395–416. [Retrieved 10 February 2014].

McQuain, W. D. (2010). Data structures and algorithms.

Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM REVIEW*, 45:167–256.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

Project, A. (1835). Dictionnaire de l'acadmie franaise. [6th Edition. Electronic version courtesy of Mark Olsen, University of Chicago].

Rossi, R. A. and Gleich, D. F. (2012). Dynamic pagerank using evolving teleportation. In *Proceedings of the 9th International Conference on Algorithms and Models for the Web Graph*, WAW'12, pages 126–137, Berlin, Heidelberg. Springer-Verlag.

School, D. (2014). Social graph— school of data, web page. [Online; accessed 2014].

touchgraph (2014). Social graph— touchgraph, webpage. [Online; accessed 2014].

Wikipedia (2014a). Correlation coefficients — wikipedia, the free encyclopedia. [Online; accessed 2014].

Wikipedia (2014b). Pagerank— wikipedia, the free encyclopedia. [Online; accessed 2014].

Wikipedia (2014c). Social graph— wikipedia, the free encyclopedia. [Online; accessed 2014].

WordNet (2014). Wordnet— wordnet, the official webpage. [Online; accessed 2014].

Appendix A

.1 Social Network analysis for Journalists using the Twitter API

The graph had 560 nodes and 1257 edges. We calculated the correlation between *PageRank* and degree of Social Network of Journalists Twitting [School (2014)](see Figure 1).

In second step, we checked what happens to the *PageRank* ranking of the nodes if we randomly omit some nodes. To examine this, we followed three steps.

1. We calculated the *PageRank* of the graph nodes and we sorted the nodes according to their *PageRank*. Figure 2 shows the ranking of first 20 nodes according to their *PageRank*.

2. We omitted 28 (5% of the nodes) random nodes by using a random numbers generator (see Figure 3). We compared the tables in figures 2 and 3, and we found that the ranking of high ranked nodes has not changed significantly.

3. We repeated step 2 by choosing and omitting a different set of 28 random nodes.

Result: We got the same result (see figures 2, 3) .

.2 Preferential Attachment

Using Netlogo we created a Preferential Attachment network and calculated the *PageRank* of this graph. Also we computed the degree of its nodes and we found the correlation between the nodes degree and their *PageRank*. The graph had 510 nodes and 508 edges. The structure and formation of this kind of graph matches the obtained negative correlation between *PageRank* and degree (see Figure 4).

By using all of the three mentioned steps for this graph, we got the same outcome. i.e., no significant changes according to *PageRank* occurred after omitting 5% of the nodes.

.3 Random Graph

We used Netlogo to form a random graph. The created graph had 500 nodes and 18814 edges which were created by probability 0.15 (see Figure 5).

Second iteration: We repeat the examination with another random graph which had 500 nodes and 12570 edges and we did not noticed any significant difference between the ranking of the nodes after omitting the 5% of them, but the correlation between *PageRank* and degree of the nodes was low.

Correlations

		PageRank	Degree
PageRank	Pearson Correlation	1	.973**
	Sig. (2-tailed)		.000
	N	560	560
Degree	Pearson Correlation	.973**	1
	Sig. (2-tailed)	.000	
	N	560	560

**. Correlation is significant at the 0.01 level (2-tailed).

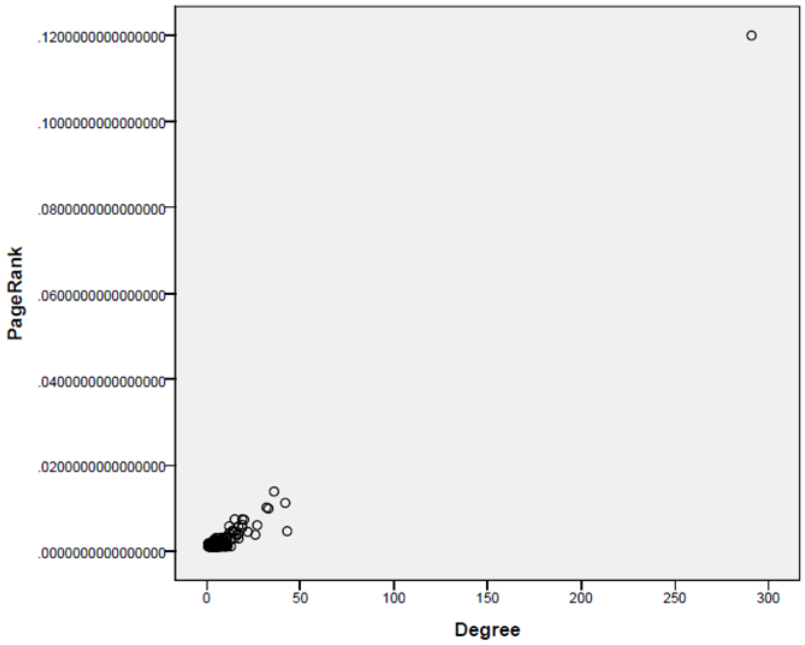

Figure 1: *Correlation between PageRank and degree of Social Network of Journalists Twitting.*

No	Id	Label	PageRank
1	124	#ddj	0.11993191
2	18	@wikileaks	0.01390717
3	416	#opendata	0.01126201
4	467	#dataviz	0.01019077
5	36	@jplusplus_	0.0099347
6	209	@odonnellmaria	0.00745491
7	316	#openspending	0.00745491
8	375	#projectk	0.00735655
9	250	#data	0.00617295
10	147	@jeanabbiateci	0.00607645
11	40	@ona	0.00583005
12	415	@momiperalta	0.00580987
13	270	#offshoreleaks	0.00548884
14	162	@icijorg	0.004852
15	382	@schoolofdata	0.00470317
16	225	@ddjournalism	0.00469072
17	130	#bigdata	0.00457869
18	64	@digiphile	0.00452241
19	23	#opengov	0.0042246
20	484	@jwyg	0.00403261

Figure 2: *PageRank of the first* 20 *high ranked nodes, Social Network of Journalists Twitting.*

No.	Id	PageRank
1	124	0.123431005
2	18	0.01368387
3	416	0.011547946
4	36	0.010131827
5	467	0.009468063
6	209	0.007902877
7	316	0.007902877
8	375	0.007128647
9	250	0.007011226
10	40	0.006180544
11	415	0.006159069
12	270	0.005822589
13	147	0.005395529
14	162	0.005147576
15	382	0.004986128
16	130	0.004975131
17	225	0.004855737
18	64	0.004796448
19	23	0.004504635
20	484	0.004275026

Figure 3: *PageRank of the first* 20 *high ranked nodes, Social Network of Journalists Twitting after omitting the* 5% *of the nodes.*

Correlations

		Degree	PageRank
Degree	Pearson Correlation	1	-.174**
	Sig. (2-tailed)		.000
	N	508	508
PageRank	Pearson Correlation	-.174**	1
	Sig. (2-tailed)	.000	
	N	508	508

**. Correlation is significant at the 0.01 level (2-tailed).

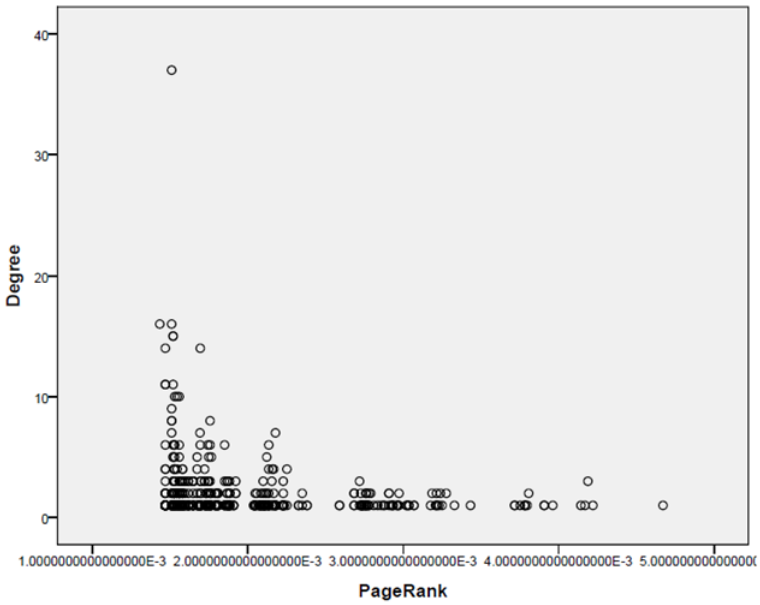

Figure 4: *Correlation between PageRank and degree of the nodes of a Preferential Attachment Graph.*

Correlations

		PageRank	Degree
PageRank	Pearson Correlation	1	.070
	Sig. (2-tailed)		.115
	N	502	502
Degree	Pearson Correlation	.070	1
	Sig. (2-tailed)	.115	
	N	502	502

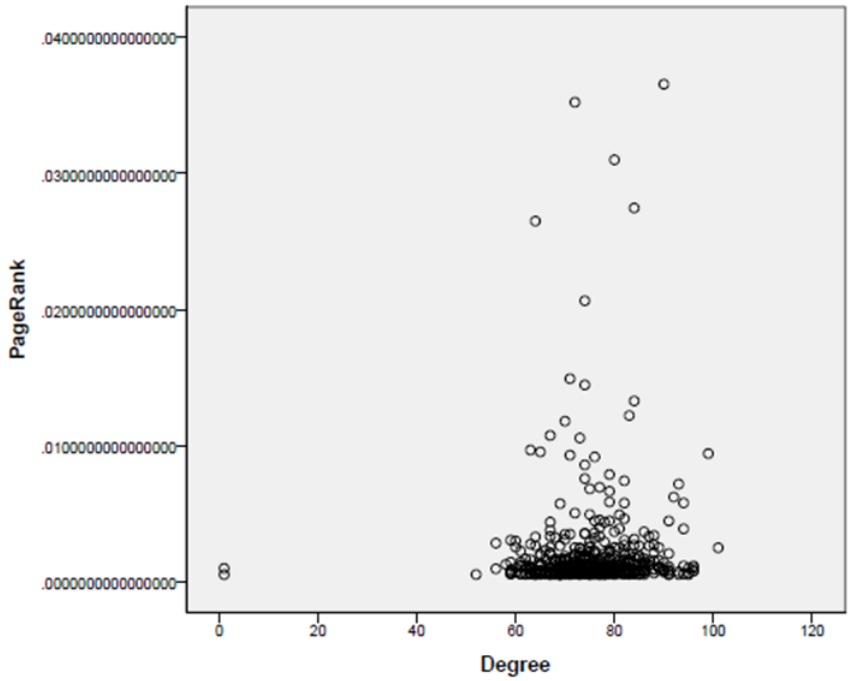

Figure 5: *Correlation between PageRank and degree of the nodes of a random graph.*

Appendix B

We created 100 different graphs which were consist of 5 nodes. The graphs were directed but non weighted. In the first step we calculated the *PageRank* of all the nodes in different graphs in addition to other properties of nodes, in-degree, out-degree, total degree and in-degree/out-degree.

In second step we omitted one of the nodes, node 3 and we calculated all of the mentioned measures, again. Then we introduced some new measures:
Δ = the difference between the *PageRank* of a node before and after omitting the node 3.
δ = Standard deviation of *PageRank* changes for every node.

The goal was finding which nodes are more resistant to the *PageRank* changes. We compared the Δ and δ with in-degree, out-degree, total degree and in-degree/out-degree of the nodes. For an example, figures 6,7,8 and 9 show the correlation between Δ and δ with different properties of node 1 . The result of researching theses graphs showed us that there is no special relation between these properties of nodes and their *PageRank* changes.

In third step, we went further. We calculated the mean of all measures for the neighbors of the nodes in all 100 different graphs.

We compared Δ and δ with new measures and we draw the charts showing the correlations, Figures 10, 11,12, 13, 14, 15, 16 and 17.

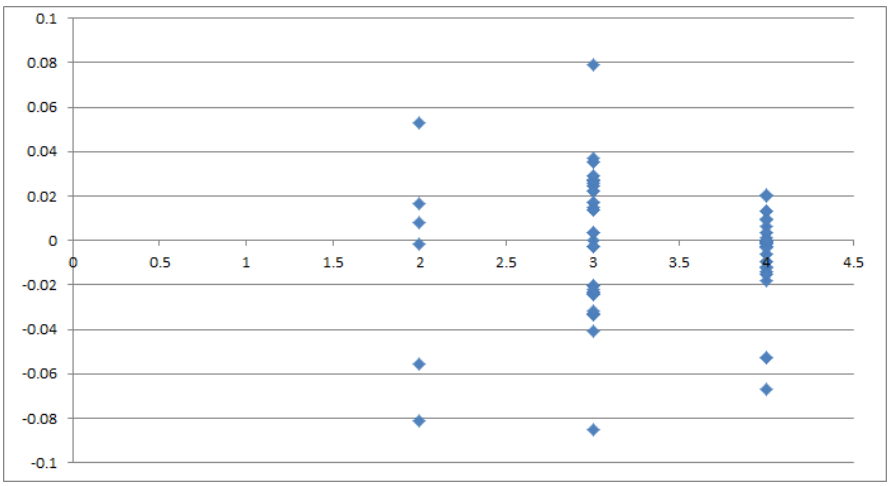

Figure 6: *Correlation between* Δ *and in-degree of node* 1.

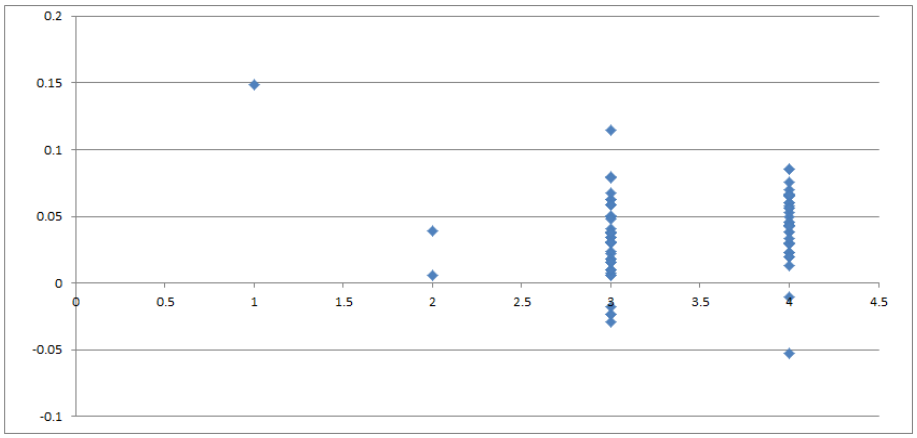

Figure 7: *Correlation between* Δ *and out-degree of node* 1.

As it is clear from the charts, at least with examining a small graph there is no obvious relation between the properties of nodes and Δ. The only obtained trend is the fact that a higher mean of a node's neighbors' PageRank causes a higher change in its *PageRank* after omitting a node.

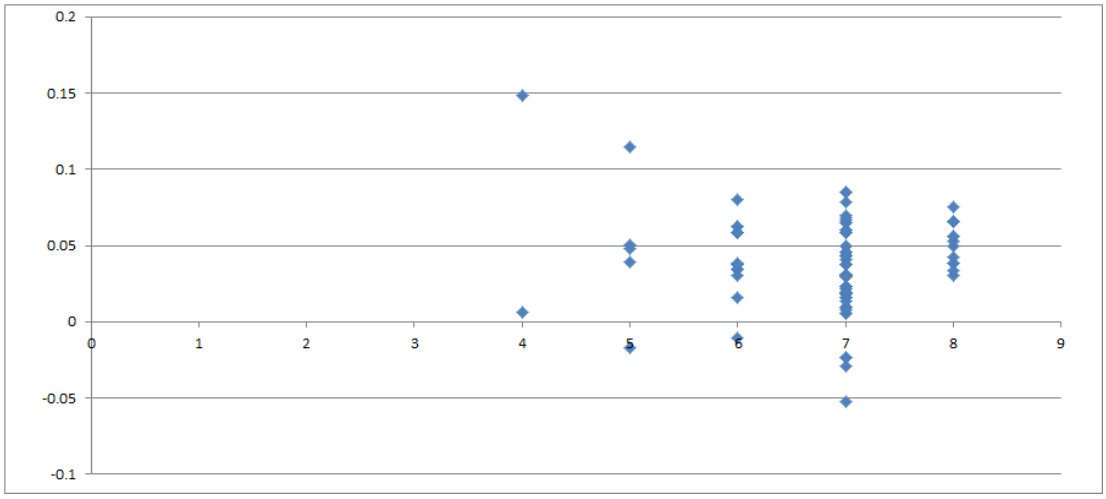

Figure 8: *Correlation between* Δ *and total degree of node* 1.

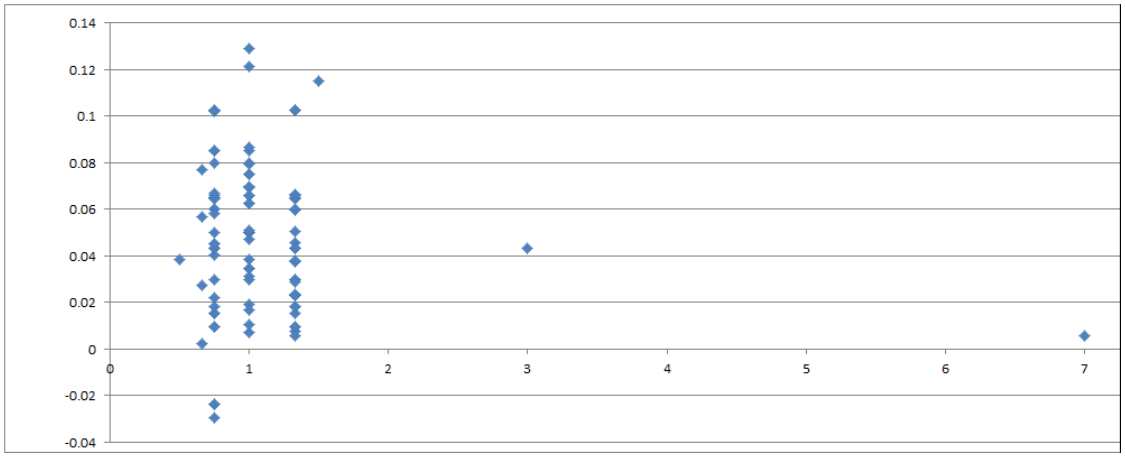

Figure 9: *Correlation between* Δ *and (in-degree / out-degree) of node* 1.

Node ID	1				
	G1				
	In-Degree	Out-Degree	Degree	PageRank	In Degree/ Out Degree
1	4	4	8	0.200000003	1
2	4	4	8	0.200000003	1
3	4	4	8	0.200000003	1
4	4	4	8	0.200000003	1
5	4	4	8	0.200000003	1

Node ID	1							
	G1							
	Neighbours	In-Degree (NM)	Out-Degree (NM)	Degree (NM)	PageRank (NM)	(In Degree/Out Degree)(NM)	Degree(A)/ Degree(NM)	In Degree(A)/Out Degree(NM)
1	2,3,4,5	4	4	8	0.2	1	1	1
2	1,3,4,5	4	4	8	0.2	1	1	1
3	1,2,4,5	4	4	8	0.2	1	1	1
4	1,2,3,5	4	4	8	0.2	1	1	1
5	1,2,3,4	4	4	8	0.2	1	1	1

Node ID	1						
	G`1						
	In-Degree	Out-Degree	Degree	PageRank	In Degree/ Out Degree	Δ	σ
1	3	3	6	0.25	1	0.049999997	0
2	3	3	6	0.25	1	0.049999997	0
3							
4	3	3	6	0.25	1	0.049999997	0
5	3	3	6	0.25	1	0.049999997	0
	Mean of Δ					0.049999997	

Δ	(G`1 nodes` Pagerank) -(G1 nodes` Pagerank)
σ	Standard deviation
G1	Graph 1
G`1	Graph 1 after omitting node 3
NM	Neighburs Mean
A	node

Figure 10: *A table of all measures for graph* 1.

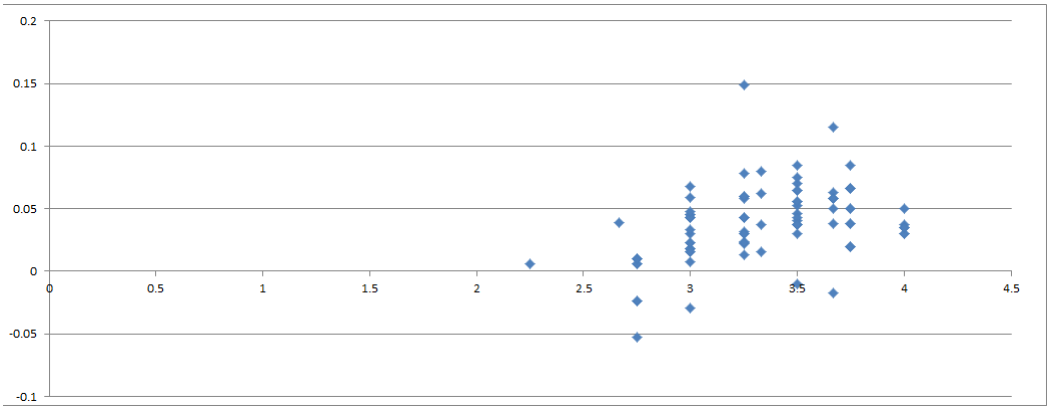

Figure 11: *Correlation between* Δ *and the mean in-degree of the neighbors of node* 1.

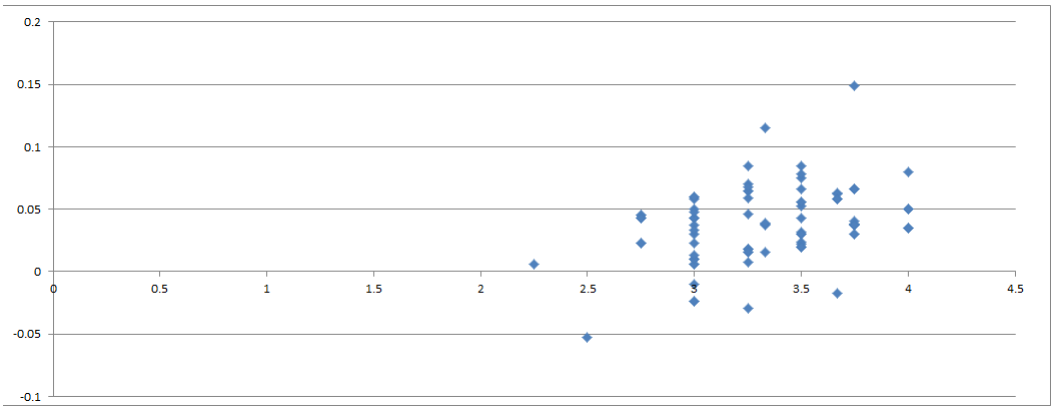

Figure 12: *Correlation between* Δ *and the mean out-degree of the neighbors of node* 1.

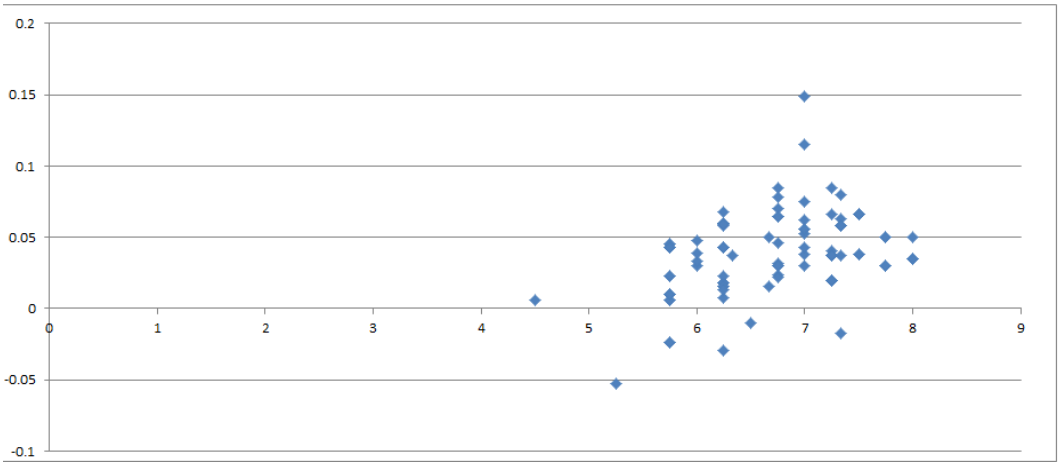

Figure 13: *Correlation between* Δ *and the mean total degree of the neighbors of node* 1.

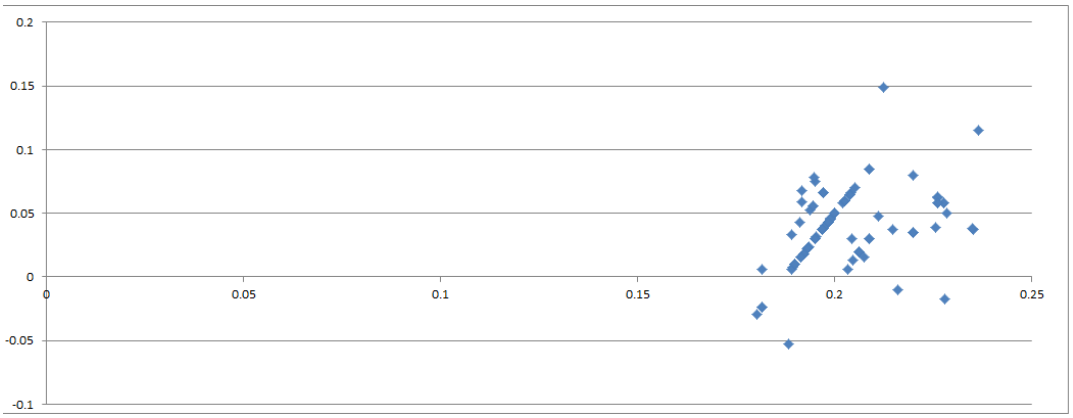

Figure 14: *Correlation between* Δ *and the mean PageRank of the neighbors of node* 1.

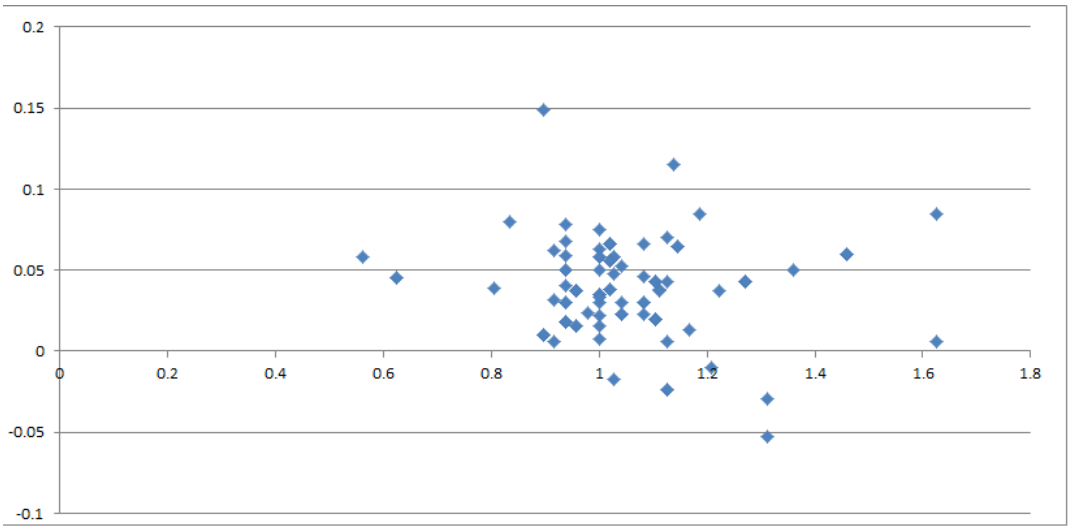

Figure 15: *Correlation between* Δ *and the mean (in-degree / out-degree) of the neighbors of node* 1.

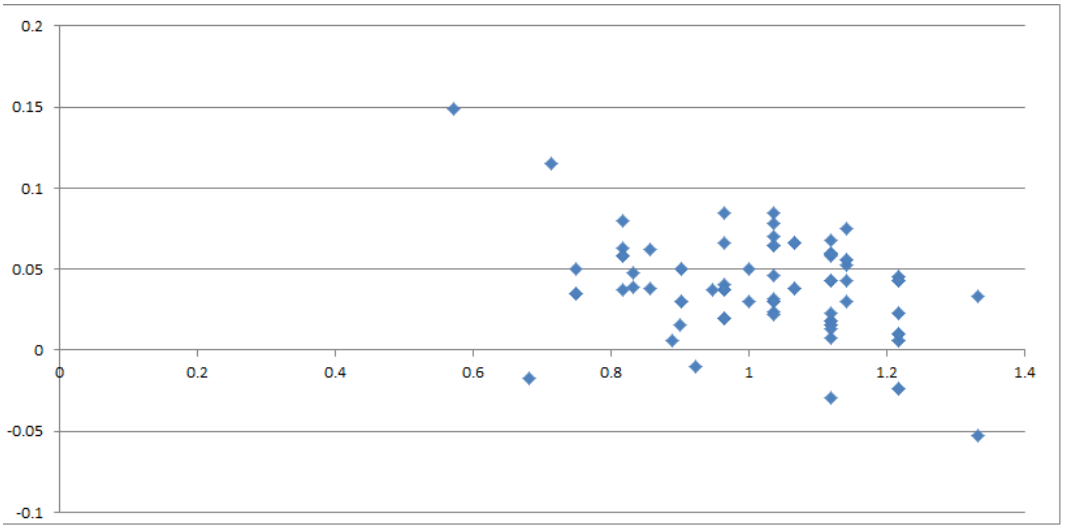

Figure 16: *Correlation between* Δ *and the (total degree of node* 1 */ mean total degree of the neighbors of node* 1*).*

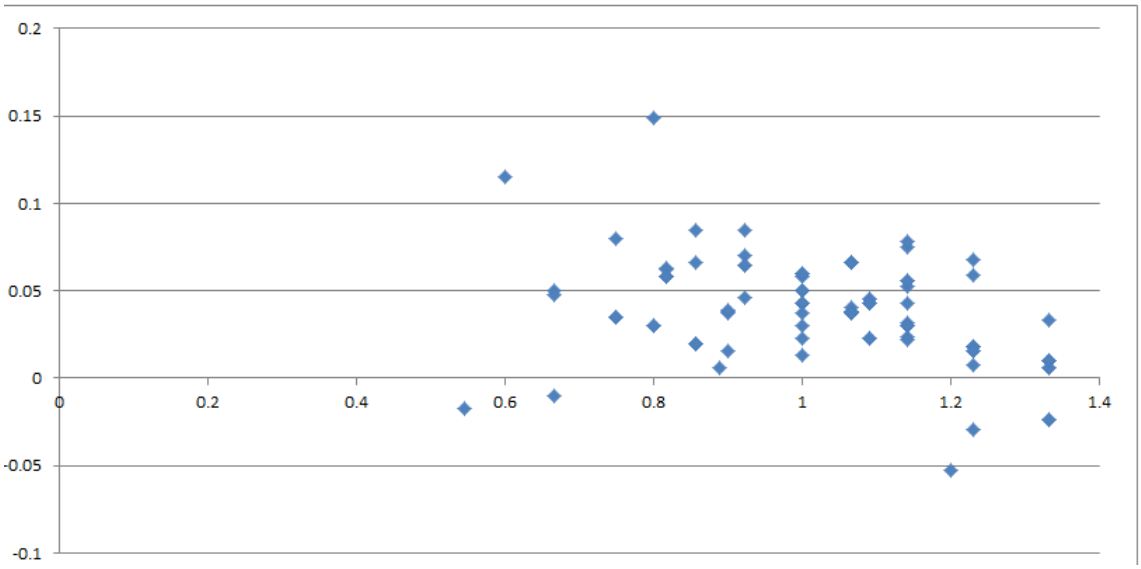

Figure 17: *Correlation between* Δ *and the (in-degree of node* 1 */ mean out-degree of the neighbors of node* 1*).*